IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 12, 2020, accepted April 29, 2020, date of publication May 7, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993191

Graph-Based Text Representation and Matching: A Review of the State of the Art and Future Challenges

AHMED HAMZA OSMAN AND OMAR MOHAMMED BARUKUB
Department of Information System, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
Corresponding author: Ahmed Hamza Osman (ahoahmad@kau.edu.sa)

This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (DF-681-830-1441).

ABSTRACT Graph-based text representation is one of the important preprocessing steps in data and text mining, Natural Language Processing (NLP), and information retrieval approaches. The graph-based methods focus on how to represent text documents in the shape of a graph to exploit the best features of their characteristics. This study reviews and lists the advantages and disadvantages of such methods employed or developed in graph-based text representations. The literature shows that some of the proposed graph-based methods suffer from a lack of representing texts in certain situations. Currently, several techniques are commonly used in graph-based text representation. However, there are still some weaknesses and shortages in these techniques and tools that significantly affect the success of graph representation and graph matching. In this review, we conduct an inclusive survey of the state of the art in graph-based text representation and learning. We provide a formal description of the problem of graph-based text representation and introduce some basic concepts. More significantly, this study proposes a new taxonomy of graph-based text representation, categorizing the existing studies based on representation characteristics and scheme techniques. In terms of the representation scheme taxonomy, we introduce four main types of conceptual graph schemes and summarize the challenges faced in each scheme. The main issues of graph representation, such as research topics and the sub-taxonomy of graph models for web documents, are introduced and categorized. This research also covers some tasks of understanding natural language processing (NLP) that depend on different types of graph structures. In addition, the graph matching taxonomy implements three main categories based on the matching approach, including structural-, semantic-, and similarity-based approaches. Moreover, a deep comparison of these approaches is discussed and reported in terms of methods and tools, the concepts of matching and locality, and the application domains that use these tools. Finally, the paper recommends seven promising future study directions in the graph-based text representation field. These recommendation points are summarized and highlighted as open problems and challenges of graph-based text representation and learning to facilitate and fill the research gaps for scientific researchers in this field.

INDEX TERMS Graph representation, NLP, graph, graph matching, representation scheme, text mining.

I. INTRODUCTION

The website has been a significant source of knowledge on every subject or domain in recent years. The amount of text generated by social media posts, forums, URLs, etc. has made it important to employ advanced methods to identify and gain valuable data patterns. Automated text recognition and natural language processing tend to be well suited for the interpretation of textual data and for the detection of relevant details in a wide variability of systems. Several attempts were made to deliver algorithms for personalized text processing e.g. the selection of subjects, text processing, etc. The effective text analysis should be emphasized that depends heavily on the way a text corpus is portrayed. Bag of Words (BOW) is a standard formalism for expressing textual knowledge defining meanings in the language (Salton *et al.*, 1975).

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman.

Several aspects are part of this representation: a repertoire of known words (the most important words generally) and a measure of their appearance. This strategy is destined to be unsuccessful, as seen in other plays, and shows a variety of unintended difficulties and vulnerabilities linked to the absence of connections. This issue therefore causes for essential problems, from both semantic interpretation and text processing perspectives. Note that as shown by Hirst [1], connections between words are of great explainable significance because their meaning is revealed in the text, thus allowing the analysis of texts to be carried out. A graph representation of text was suggested as an solution to solve the shortcomings of BOW approaches to cope with this issue Wang *et al.* 2011 [2]; Jin and Srihari [3]; Zhou *et al.* [4]; Rousseau and Vazirgiannis [5]. The above have been researched primarily as a way to take time dependency and term orders into account. The co-occurrence network, one of the most popular text representation formalisms and has been implemented in various modern systems. In comparison to the BOW model, this model provides an essential context to describe relationships among words. A text is basically represented as a graph where vertices display coincidences of words and edges. In the literature a variety of versions of the standard representation of co-occurrences is suggested. For eg, in Sihag and Kumar [6], the initial centroid parameters for the K-means algorithm were evaluated by a co-occurrence network. In Hossain and Angryk [7], authors suggest to use the WordNet [8] lexical basic information to first generate document graphs, then use them for category and text analysis.

During the Big Data era, text is one of the most omnipresent processing types. Data representation is an essential step in the data mining feature extraction process. Therefore, there is an ongoing challenge in determining a correct model for text representation that can considerably capture the inherent features of textual data. New models receive high appreciation because of the simplicity and shortcomings of traditional models such as the vector space model. Words are loosely arranged in clauses, phrases, and paragraphs to explain the meaning of a text document. Additionally, it is important and useful to understand the document in-depth, to structure it and to determine its location and the relationship between various components of the document. Text representation based on graphs can be recognized as one of the genuine solutions to the above-listed shortcomings. A text document can be viewed in many ways as a graph. In a graph-based scheme, nodes represent the characteristics and boundaries of various nodes. Whilst many graph models exist [9], a co-occurrence word graph is a good way to represent a relationship between one phrase and another in the context of social media such as Twitter or short text messages.

Currently, text is the most public form of information storage. Document representations are a significant stage in the text mining procedure. Therefore, the challenging task is the correct representation of the textual data that will be capable of representing the text's semantic information. Traditional models such as the vector space model consider numerical vectors in a Euclidean space, and latent semantic indexing (LSI) is applied to the text vector to decrease the dimensional space by correction analysis construction of the terms in collections of documents. The VSM is commonly known as the bag of words (BOW) model, and it is the standard model for document representation. The main disadvantage of the VSM is that it is impossible for the SVM to express the essence of a text and structure. Furthermore, words are independent of each other; it is not possible to represent a word appearance sequence or other relationships. Moreover, when two documents have identical definitions but different words, similarity cannot be easily determined. To describe the meaning of the text, the terms are structured into sections, sentences, paragraphs, and phrases. Therefore, it is important to understand the relationship between various document components, their ordering, and their place in detail. One of the best solutions to these problems is the graph-based text representation method [10]. Representing text as a graph is a computational construct that can effectively model the relationship and structure of data. Text reported in a graph representation is important because it can be used in most text operations such as those that are topological, relational, numerical, etc. In this research, different methods are discussed for modelling text documents using a graph. This study also discusses various methods of text document analysis based on graphs. LSI is a technique that is applied to a text vector to decrease the dimensional space by correction analysis construction of the terms in collections of documents. It is generally used in information retrieval fields. The TF/IDF algorithm is usually combined with the BOW approach in text clustering or classification in text mining. This study surveyed some of the key methods for graph-based text representation and graph matching. Through this survey, we found several limitations and advantages for those methods. The article serves as an invitation to the graph representation researchers to solve these limitations. The remainder of the study is ordered as follows: The second section discusses an explanation of the graph-based representation of a document. The third section discusses graph matching techniques, and the fourth section concludes the study.

II. TEXT REPRESENTATION SCHEMES

A map G is a fourfold graph: $G = (V; E; \alpha; \beta)$, where V is a set of vertices, and $E \subseteq V \times V$ is a set of graph-edges with lines connecting the vertices, $\alpha : V ! Lv$, $\beta : V \times V !$. The labelling functions of the vertices and the edges are the labelling functions (the labelling sets will appear on the vertices and edges, with Lv and Le). By omitting the labelling functions, we may refer to G as $G = (V, E)$. A graph $G1 = (V1; E1; \alpha 1; \beta 1)$ is a subgraph of a graph $G2 = (V2; E2; \alpha 2; \beta 2)$, indicated by $G1 \subseteq G2$, if $V1 \subseteq V2$, $E1 \subseteq E2 \setminus (V1 \times V1)$, $\alpha 1(x) = \alpha 2(x)\ 8x\ 2\ V1$, and $\beta 1(x; y) = \beta 2(x; y)\ 8(x; y)\ 2\ E1$. Equally, graph $G2$ is called a supergraph of $G1$.

Several graphs are available. An undirected graph is one with no orientation on the edges. The edge (a, b) is the same as the edge (b, a). In addition, a graph with the directed

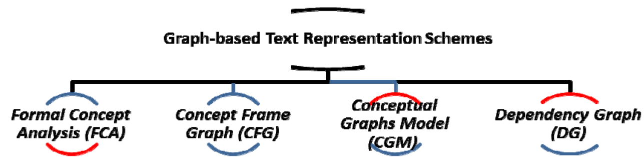


FIGURE 1. Types of graph-based text representation schemes.

edges is called a directed graph, or digraph. In addition, the concept of a multi-graph refers to a multi-graph that requires multiple edges between nodes. An additional common graph category is called a weighted-graph, which is a graph in which each edge has a linked mathematical value, termed as the weight. Typically, the edge-weights are non-negative integers. Weighted-graphs can be either undirected or directed. The suggested web document graph models include several that represent content for web documents (and generally text documents) as graphs and were proposed by [11]. They also suggested a variety of distance measures and similarity measures between graphs for text classification and reported substantial improvements in document classification accuracy with the graphical approach versus a bag of words. Nonetheless, these graphs revealed that running algorithms are much slower. The graphical representation problem also resides in the lack of model-based classifications for documents represented by graphs [12].

Some authors suggest using frequent subgraph mining to create an integrated model to address these problems [12]–[14]. Frequent mining of subgraphs is used in this method to find a list of subgraphs between graphs representing text documents. Subsequently, these subgraphs may be considered as a word, as in VSM; then, documents are represented as a vector of word weights.

Previous research attempted to show the contents of the text using graphical schemes, such as the dependency graph (DG), formal concept analysis (FCA), concept frame graph (CFG), and conceptual graphs (CGs). Figure 1 presents the types of text representation schemes. Figure 1 shows the main types of graph-based text representation schemes.

For the outcome of the cluster, the variable is very important, and it is essential to choose an appropriate model for the representation of the abovementioned text models. In general, the texts of this study are represented by using graph types, and details of each type are shown later, where words represent the correlation between the words as a node in a graph or between the two nodes (edges). This result shows that improved mining can be achieved by graphically representing document information. The application of the stem algorithms, lemmas, etc. must be the first step to determine the terms in the document. With stems or other techniques, each word shown in a document becomes a graph node to normalize language-specific algorithms. At that point, each node in the graph is unique because each node has its term, and even when the same term is repeated in one document, it is also considered unique.

The second task is to find a coordinated edge between the nodes of the term ***A*** and the node with the term B with the edge mark ***B***. If a word ***B*** indicates a place in an "rea" of the content substance, title, or connection, etc., then ***S*** of the document follows. An edge cannot be made between two words given the possibility that certain punctuations have been isolated [15]. The graph will capture basic content information (site and relative place of the word) with the present representation. The format consists of three parts, including the name, reference, and text. The title includes the archive title and any keywords (metadata) provided. The anchor text that appears in the document is called the connection in hyperlinks. The text involves the document content (this includes hyperlinked contents, but not the titles and keywords of the document).

A. FORMAL CONCEPT ANALYSIS (FCA)

Over the past decade, a wide range of application fields in the international community have been developed, for example, psychology, AI, data, and data analysis, and some specialists use other kinds of graphs in a text representation; in particular, "formal concept analysis" (FCA) was recently enhanced by [16] and [17]. FCA is the basic method used for an arrangement of objects and properties in a hierarchy or formal ontology. FCA is the fundamental method used for an arrangement of items and features in the concept of hierarchy or formal ontology. Each concept is represented in the hierarchy as a collection of objects that share similar properties for a certain group of properties. In the ideas above, the sub-concept in the hierarchy includes a subset of posts. The technique was derived from Garrett Birkhoff's application of the lattice and order hypothesis in the 1930s and includes information from the analysis to clarify the conceptual structures of the dataset. In FCA, the measure of similarity depends on "Tversky's model". The items in FCA are referred to as "formal objects", which are also known as the "formal property" items as an alternate type of description. The "formal" adjective is used to validate the formal definition. Formal objects do not always have to be "objects" in any logical sense of the "object." In many cases, it is, however, useful to choose object-like elements as formal items and components or properties as formal features in the use of "objects" and "attributes." However, this sign is given in FCA. Information is analysed and knowledge and data management are represented by [18]–[20]. In addition, an FCA-based approach has been developed to break down the data sparsity effect of an adaptive model. Documentation may be treated as 'object-like' when retrieving data, whereas the words may be seen as 'attribute-like' [21]. Furthermore, elements, such as tokens and the kinds of things, qualities and information (information that is driving news and speculation, words and implications, and so on), comprise a group of formal elements and their formal qualities [22]. FCA has practical applications in the area of data mining, content mining, apprenticeship management, learning administration machine learning, programming development, research, semantic web, etc [23].

FCA uses further analysis by providing a method to boost IR in light of the FCA website. Semantic connections are built by questions and allow ideas to be updated in a window. The replies will then be assembled using a web index [24].

B. CONCEPT FRAME GRAPH (CFG)

In the text representation, the analysts use some sort of graph. Several authors have suggested a method for training to build CFG data from the contents of texts. In addition, the CFG is based on conceptual knowledge and data creation by the basic structural architecture to address the question of the material with the definition. Consequently, a new technique known as the concept frame graph was created. In a customer-oriented knowledge sharing scenario, an intuitive concept description framework is implemented from the learning base. During empirical studies, researchers found that the suggested method is a promising approach to obtain more data from credible documents and the realities of life [25]. Rajaraman and Tan [26] analysed mining execution with and without graph-based text representation. Algorithms that were not effective in the use of other graphic approaches relative to the CFG method obtained improvements in precision and recall of 35% and 18%, respectively. Preprocessing steps, such as stemming, lemmas, etc., must be defined first to determine the words in the text. With the stemming algorithm or with other methods, each term in a document becomes a node in the graph to normalize a language algorithm. All nodes in the graph are unique and distinctive since every node has its term, even when the same term is repeated in a single document. The second task is to coordinate the edge from the node of the term A to the node, compared to the term B, with the edge mark B, if a word B is immediately placed in the ''area'' (substance, title or connection, etc.) after the word A. There is no difference between two terms regarding the possibility that certain punctuations have been separated [15]. The graph can record basic information of the content (place and place of the word) with the current representation.

C. CONCEPTUAL GRAPHS MODEL (CGM)

The third type of diagrams we present are ''Conceptual Graphs'' (CG), as discussed in Sowa and Way [27], which indicates that the ''Conceptual Graphical Model'' (CGM) is more capable of understanding. Montesy-Gómez *et al* [28] and [29] are experts who stressed the use of this type of graph for the extraction of text features or classification work with language for representation of knowledge. The approach is well known in psychology, philosophy, and linguistics. The information structure at the semantic level could be expressed in CGS. The CGs are therefore bipartite, connected and tight. A diagram contains an array of edges and vertical nodes. The CGs distinguish between the relationships of any arity and anything remaining in the dialect of a system using a circular segment. The CGs are similar to diagrams used in the usual dialects. CGs can address accurately and deeply organized data. A built CG is often regularly used for graph planning; it produces results that are accurate for various purposes. In the technique of viewing knowledge in text, the contents of the document are viewed with the CG formalism and the CG match is performed. The CG has been used to document the usual structure of the text through various works in [30] and [31]. Most of our works take the linguistic structure of the contents as a basis for parsing projects before transformation into CGs. In this exam, CG work to effectively track the semantics and structure of the extracted data given their ability.

In hospitals, the conceptual graphs are used to obtain free text in the medical document and acquire semantic data and information. The employed software and auto sorting methods are used to develop combining principles from generic medical classifications and extensive arrangements of clinical repositories for free-content [23]. For ordering ''Extensible Markup Language (XML)'' files, [32] used the CG representation. The data are installed in the archive as a metatag. This method incorporated two phases; the concept of semantic parts was then used to create specific CGs with the data. Similarly, the projection algorithm focuses on the basic resemblance among CGs, and the best time for implementation is NP. In the work of Abdulsahib [33], graphs were built with a view to two proposals; in a phrase, we have a relationship among the words within a modulo frame estimate of the ideal size of six (when the separation between terms is equal to or lower than six tokens, the edges are created). A few reviews focus on a robotized thematic that can provide customers with the benefits of separating and understanding the accumulations of reports, as well as web indexes that focus on the relationship between word collections and their latent topics. Nevertheless, the present approach to this ensures quality by focusing on the structure in the data mode. The findings were drawn from [34]. It was found that by using the graph approach, the ideas that represent the best topics could be classified. The advantage of this type of graph captures the relationship between terms. However, the drawbacks of these kinds of charts are the arithmetical complexity in comparing graphs. One drawback of this (CG) approach is that it becomes distinctly polynomial and has a wide range of parameters. There are some methodologies for using a full content representation, not just words and basic relationships between words. Conceptual graphs (CGs), as exhibited in the template proposed by [27], are one of the standard methods for capturing semantic connections between languages. In CGs, the concepts and relationships exist in two types of nodes. The semantic part of the episode ideas is shown in a relationship node.

By interpreting CGs to predicate analysis, a semantic significance of a sentence can be gained. The ISO/IEC 24707 Standard for common logic that characterizes semantics in terms of dynamic linguistic structure and model theoretical semantics is the official standard for conceptual graph linguistic structure and semantics. Nevertheless, the meaning of natural languages is difficult to change to the systems of the CGs [35]. Most works can be divided into manual development, deterministic methods and observable methodologies in the building of CGs. For example, [32] portray semi-automatic conceptual graphical text presentations using a mixture of existing language resources, such as VerbNet and WordNet. The main idea of this strategy is that VerbNet and WordNet were used by the creators to distinguish semantic parts. All records have been converted to XML format in the first instance. They used a syntactic parser to search each phrase and recognize sections using VerbNet at that stage. The principal verb was distinguished from each proviso in the sentence, and a sentence example using the parse tree was built. All imaginable semantic edges from VerbNet in every verb in the sentence were removed. Finally, each sentence was drawn up in the concept graph using standard CG principles [32]. Ordoñez-Salinas and Gelbukh [35] proposed a linguistic utilization to be used in light of the dependence and the standard characteristic of conceptual graphs. The scientists used noun pre-modifiers and noun post-modifiers and verb contours separated from VerbNet to produce the grams of dependence, which include verb classification, their syntactic portrayal, and framework depictions, as a source of the meanings of semantic components. The sentence is designed to resemble CGs for the constructed trees [32]. To summarize, rich semantic material information can be captured in a graph through the use of CGs, but the fact remains that creating such a plot is not an easy task.

D. DEPENDENCY GRAPH (DG)

The latter kind of graph in this analysis is a "dependency graph". The dependency graph shows the dependencies of many items in a coordinated graph. DG is a type of content representation scheme that linguistically characterizes the form of a sentence, which demonstrates how distinctive words associate through direct connections called dependencies. The current approach has enabled dependence on the modelling of words, terms or whole words. It is possible to have a decision regarding whether or not an association is considered to complement the graph [36]. This graph represents the relationship between dependence accurately. This graph is an independent language, which means that it can be used in any language for text normalization. The graph contains a set of proposals (nodes), an assertive use of nodes and a sequence of dependency connections (connecting the brackets), which limit the secrets of waiving. Privacy is decided as entirely (one value), a specific part (many values), or an unknown part (all values). Such graphs concentrate on causal links among the words and improve the quality of the measurement of similarity among the texts [2]. A dependence chart is defined. The coordinated graph is expressed as "G = (V, E)" by [37], where V represents arrangements of nodes (pairs) and "E = VV alternative" is the edge arrangement (conditions). We will check the previous reviews in the text representation of the used dependency graph. The object dependency exploration model (ODEM) was applied in [38]. The graph of dependency encrypted in "ODEM" contains groups as nodes of the actress. These nodes have an explanation regarding how they are classified, such as class, interface, explanation, reflection, finalization, and vision. Each node contains a list of relationships (dependencies) in one direction, and the full class name (packageName.className) and explanations for the classification of dependencies are also provided. This process improves perception and thus shows that it is much less complicated to look at the graph. The researchers [23] suggest a novel "FEDG" model that can provide more effective data compared to the CG model. FEDG is a new model that offers better details. Furthermore, a new clustering method has been introduced that combines dynamic research and static dependencies. The dependency graph provides a reference representation of fundamental relations between the classes. A graph is at the latest directed diagram of two edges between two groups. A programmed undertaking supported by various tools is used to extract structural relations. In their support of various innovations, extractive devices vary [39]. Some experts suggested a diagram-based approach using a two-area graph display (site pages and email) that included separate graphs [40]. The graph representations are selected based on field knowledge to highlight the various fields. During the same year, both authors used the source code review system for the development of the dependency diagram describing framework modules and the module level between relationships [41]. This graph then used the bunching method, which segments the graph as a point of entry. The results were presented using graph visualization in a clustering graph. The algorithms that depend on the graph showed that the experimental results obtained by [2] were better in a specific document of methods based on the BOW model. This approach can also define causal relationships and improve the execution of the textual similarity steps. Beck and Diehl [42] found a new approach that involved the integration of dependency graphs before the clustering was carried out and the associated arrangements for operations such as "union, weighted union and a group of edges intersection". The authors concluded that the application of the two methodologies increases the essential reliability of the clustering. Table 1 represents the comparison between the types of graph-based text representation schemes.

III. TAXONOMY GRAPH-BASED REPRESENTATION

Graphs can be used to represent different relationships (e.g., words, persons, sentences, and documents) among different semantic units. Graphs are general data structures that represent complex relationships between different entities. Several topics, knowledge methods, and techniques in information retrieval (IR) were proposed for representing text documents as graphs. These methods can be classified, as shown in figure 2.

In this survey, examples of studies of topics and applications that were used and applied to the graph-based representation in the different research fields have been reviewed and discussed. It makes it hard for Internet users to retrieve the most relevant information on a specific topic quickly by this digital information explosion. Several topics were

TABLE 1. Comparison between the main types of the graph text representation schemes.

Author	Graph Scheme	Advantages	Disadvantages	Ref
Valatkaite and Vasilecas	CGM	Conceptual IS-graph is versatile and accessible at various levels of information system development activity.	CGM is an un-ordered set of concepts and supports only data	[43]
Wang *et al.*	DG	Detect causal relations and improve the performance of a textual equivalence measurement.	Obligatory to enhance the texts visualizations	[2]
Wang and Liu	FCA	ability to detect the relationship between terms	Complex in computational time Compared with other graphs scheme	[23]
Rajarama n and Tan	CFG	Provides a description about, targets, semantics frames, and semantic role for each term in a text using Frame-Net resource linguistic and theory of Frame Semantic.	Shallow of semantic information	[25, 26]

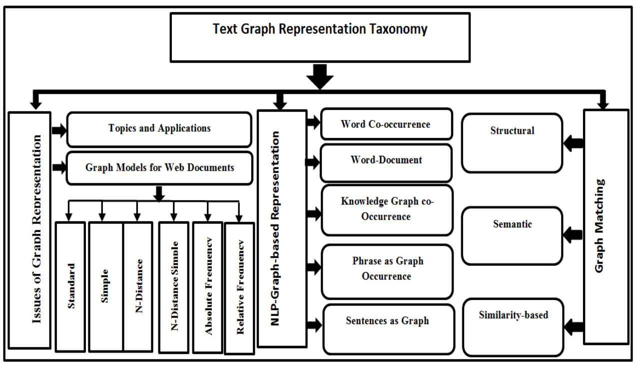


FIGURE 2. Taxonomy of graph-based representation.

discussed, and research studies were conducted and reported for representing and applying the text as a graph.

A. TOPICS AND APPLICATIONS USING GRAPH- AND SUBGRAPH-BASED REPRESENTATION

Many graph-based representation approaches were introduced and adopted to solve a semantic plagiarism detection problem [44]–[47]. Graph-based representation was adopted to solve a plagiarism detection problem. The proposed method represents each sentence inside a text document as a form of a node and combines all the terms of the sentences in a node. The concluded nodes are linked to each other according to the sentence order inside the text document. The extracted nodes are then coupled with one large node at the top level called the topic signature node (TS). The comparison between the graphs was done based on the topic signature nodes.

In the semantic role labelling (SRL) approach, the model defined in [48] also increases the graph representation with Propbank-style semantic roles. Each predicate adds the head of the argument phrase as a term role with the correct semantic position such as subject, object, verb, etc. This helps to connect words that share a profound semantical connection, which is not apparent in the surface syntax.

Sentences are classified based on the frequency of words and the frequency of sentences [49]. The sentences included are selected for summary sentences after removal of the stop word and stemming from the high-frequency word. Summaries of the high rating phrases are selected. A summary of the same topic or context is provided. Duplication of summary sentences is the main drawback of this process. For sentence extraction, as the document name, the first and last sentences of a document or each article are considered by [50], suggesting a straightforward approach. He argued that the first sentences of newspaper articles present a substantial opportunity for summary inclusion. However, the last paragraph and final parts are very likely to be outlined in technical papers. Lin and Hovy [51] maintained that the place method of Baxendale is not appropriate for the extraction of sentences in various fields. A sentence's speech structure varies from one domain to another. This system's main disadvantage was domain-related. Edmundson [52] suggested four parameters to extract the summary text. The approaches are location, keywords, cue phrases, and title words. The main disadvantage of this method was repetition in the text summary. Barzilay and Elhadad [53] proposed an approach for summarizing the sentences based on the lexical chain method. In [54], the lexical chain concept was introduced. In the various sections of the document, the lexical chain links the semantic terms. For building lexical chains, [53] used WordNet.

El-Said *et al.*, 2015 proposed to establish an efficient methodology for organizing and presenting graphic texts based on semantic annotation and Q-learning [55]. This methodology is based on semantic concepts that represent the text in the document, detect unknown dependencies and relationships between concepts in a text, measure the relationship between text documents and use the representational and relativity measures to implement mining processes. The programme reflects the current relations between concepts and provides precise measurements of the interactions that lead to better mining efficiency.

Several research projects have employed graphical representational methods for sentiment analysis such as [37]. Text corpus is known as a marked guided graph with words as nodes, while edges indicate the syntactic relationship between words. They proposed a new path constrained graph walking approach where high-level information about important sequences directs the process of graph walking. We have shown improved performance and scalability by the graph walking algorithm. The word-graph sentiment analysis method was similarly introduced by [56]. In the model, a well-described graph structure was suggested, alongside a variety of graph similarity approaches. The model extracts vectors for use in the classification of polarity. In addition, a graph-based semi-supervised algorithm was proposed in [22] to achieve a sentiment classification by solving an optimizer problem.

Peng *et al.* [57] introduced a new CNN ongoing, large-scale multi-label text labeling system, hierarchical taxonomy recognition and focus graphic capsule. The method was initially used to represent each document as the word order and normalize it as a matrix representation which preserves both the sequential seminal sequences of the non-, long- and local semiconduct. The term matrix is then applied to the planned repeating CNNs of the focus capsule to understand the semantic functions more efficiently. The hierarchical method of embedding taxonomy has been introduced to learn their representations and to establish a new weighted margin loss by the use of similarity in label representation in order to reinforce the Hierarchical relations between class labels. The model increased the performance of large-scale multi-label text labeling considerably.

Recommendation systems notify users of specific products and data based on different types of information, such as users' past shopping and product features, by predicting the interest of users in an item. Huang *et al.* 2002 used a graph-based representation method for the digital library [58]. The study commented on how they tested the concept of using a visual model of suggestions, which incorporates content-based and collaborative methods. Due to the similarity of their problem with a concept recovery project, the high-grade database, client and library associations were exploited via a Hopfield net algorithm. To evaluate the system, it has been established that the system is improved both by precision and recall by combining content-based with collaborative approaches, sample holdout testing and the preliminary subject test. Yang and Toni 2018 introduced a visual recommendation system that learns and utilizes user space geometry to create meaningful clusters in the user domain [56]. In the context of book recommendation from generic to content-based, collaborative or hybrid approaches, the two-layer graphic model was defined. A suggestion is a graph search operation using their template, and different approaches to graph search can be applied. This reduces the dimensionality of the problem while maintaining the exactness of MAB. The study then evaluates the effect on MAB quality of graph sparsity and cluster sizes and generates exhaustive simulation results both in synthetic and in real-world datasets Yang and *et al.*, 2018. Jang and *et al.* 2017 suggested a recommendation system based on a graph to record embedded similarities among items not directly connected to them. The research was seen as an alternative to traditional models as a step in the path [59]. The RERA recommender system implemented by [60] used an updated NELL information graph consisting of entities and relationships to recommend content to the users. RERA describes the user-intensive NELL entities and NELL entities listed in the content proposed. To determine how well-related the content of these units is to ranking the importance of the proposed data, RERA used a new, improved page ranking algorithm.

Graphs are not just useful as organized knowledge repositories. In modern machine learning, they also play a key role. Apart from graphical structured information, machine learning applications are designed to predict new patterns. For example, one may want a biological interaction graph to classify the role of a protein [61], predict a person's role within a collaborative network, suggest new users in a social network [62], or foresee a new therapeutic application of current drug molecules, the structure of which can be represented as a graph [63].

For visualization, clustering, classification of the nodes and prediction of the links, the most popular cases are node embeddings, and each of these uses is relevant to some application areas from computational social science to computational biology. In the discovery of patterns and visualization, a long history is presented with the problem of viewing graphs in the 2D interface and applications in data mining, social sciences, and biology [64]. Node embedding delivers a powerful new visualization method, which means that researchers can readily use generous techniques to visualize high-dimensional datasets as nodes are mapped to robust vectors [64], [65]. To produce 2D views of graphs [66], [67] that can be helpful to find communities and other hidden structures, for example, node integrations can also be combined with well-known techniques such as t-SNE [64] or principal component analysis (PCA). Likewise, node integrations are a powerful tool for the clustering related nodes, which has many applications from computational biology (e.g., drugs) to advertising (e.g., finding associated products) in a similar vein as visualization [68]. Again, because each node is connected to real-world vector integration, a standard clustering algorithm can be used for the collection of learned node embedding. Again, since every node is related to actual vector embedding, any generic cluster algorithm (k-means, DB-scan, etc.) can be applied to the set of learned node embeddings. This application provides an open and powerful alternative to traditional community detection techniques and provides new methodological opportunities since node embedding systems can capture functional or structural roles, not merely community structure, played by different nodes. Node classification may be the most common benchmarking method for node embedding evaluation. In many instances, the classification function is a semi-supervised learning process, in which labels only exist on a small number of nodes to label the entire graph based on this small initial seed set. Popular applications of semi-supervised node classification include the biological classification of proteins [69] and the categories of papers, images, web pages or individuals [69], [70].

The inductive node classification task of [61] has recently been introduced to classify nodes that have not been seen during the training, for instance, classification of new materials in evolving graphs of information, or generalization into invisible protein-protein networks. Node embeddings are also extremely useful as link prediction features where there are missing edges or edges are to be predicted for future formation [62]. The link prediction is at the heart of advisory systems and common node embedding applications reflect this deep connection, including the prediction of the failure of social network friendship links [67] and user/film affinities [71]. Additionally, in computational biology, the relation prediction has important applications. Many graphs of biological interactions (e.g., between proteins and others, medicines and diseases) are incomplete as data derived from expensive laboratory experiments are relied upon. Links in these noisy graphs are an important method for expanding biological datasets automatically and recommending new wet laboratory experimentation directions [72]. More generally, connection prediction is closely linked to relative statistical learning [73], where missing relationships between entities can be predicted in a knowledge graph [74].

The framework for text classification using graph convolutional networks had been suggested by Yao, Liang and *et.al.* [75]. The approach creates a composite text network with a composite corpus based on the word co-occurrence and the word associations in documents and then discovers a Text GCN. It was initialized by a single hot word and paper representation and learns the embedding processes for both words and documents, supervised by the known content type labels. The tests of the program revealed that the language GCN was better than the other classification methods. Similarity, Zhang *et al.* [76] suggested a heterogeneous graph neural network model called the HetGNN model to represent the heterogeneous conceptual structure. The approach used a random walk with a restart technique for checking for every node and grouping them based on node forms of a fixed size of closely linked heterogeneous nearest neighbors. They then developed a two-module neural network architecture to combine the function details of the neighboring nodes sampled. The first module codes ''huge'' features heterogeneous content interactions and includes object embedding for each node. The second module aggregates contents (attribute) of embedding various neighboring groups and blends them in order to achieve the optimal node embedding by taking into account the results of different groups. Lastly, mini-batch descent technique and graph context loss used to train the end-to-end pattern. In many graphical mining tasks, such as relation estimation, suggestion, node classification and clustering and inductive node classification and clustering, HetGNN proved outperforming the current baselines.

Bai *et al.* [77] introduced a new solution to this classic but challenging graphic problem, focused on a neural network, aiming to reduce computational burdens while retaining good efficiency. Two methods incorporate the suggested strategy, called SimGNN. They implemented a learning embedding method that maps each graph onto a built-in matrix, providing a description of a graph globally. In order to highlight essential nodes in relation to a particular parallel metric a new method is introduced. The method for a comparison of a pair of nodes was developed in order to complement the graphical integration of fine seeds of nodes. They argued that their model generalizes best on the unseen graphs and operates in quadratic time relative to the number of nodes in two graphs, in the worst cases.

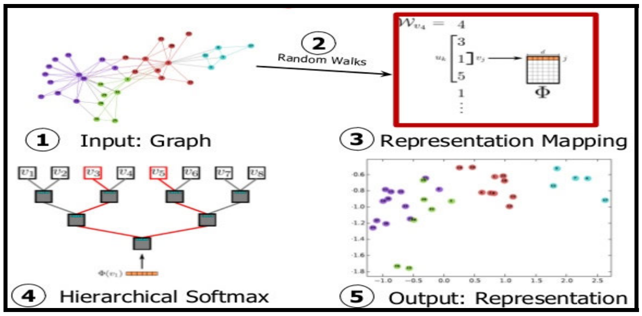


FIGURE 3. DeepWalk representation [66].

Recent progress on graph representation learning is based on unsupervised node representation, semi-supervised node representation, and learning representation of the entire graph. The graph can be preserved based on the similarity between the nodes such as DeepWalk [66] and LINE [67]. DeepWalk is a novel approach to the latent representation in a network of vertices. Such latent representations cover social relations in the continuous vector setting that statistical models can easily exploit. DeepWalk generalizes recent developments in language modelling and unsupervised function learning (or deep-learning) from word to graph sequences. DeepWalk uses local data from truncated random walking to learn latent representations by treating walks as sentence equivalents. Social representations are latent characteristics of vertices that capture the similarity and membership of the community [66]. It generalizes neural language models to process a special language composed of random walks. The semantic and syntactic structure of human languages [78] and logical analogies [79] were used for these neural language modelling approaches. Figure 3 below demonstrates the DeepWalk representation.

Large-scale information network embeddings (LINE) [79] is another successful, non-random-based approach, and the contemporaneous approach to direct coding is the LINE method [61], frequently compared to DeepWalk and node2vec. LINE combines two objectives of encoder decoders to optimize the proximity of the ''first-order'' and the ''second-order'' graph. The first-order target uses a sigmoid-based decoder and proximity measure of graph adjacency. The encoder-decoder of the second order is identical but takes into account two-hop neighbourhoods adjacent to it. The goals of the first and second orders were configured using KL divergence metric loss functions [79]. LINE thus has a conceptual link to node2vec and DeepWalk in that it uses a decoder and lacks probability but it specifically factorizes first- and second-order proximities rather than combining them in random walks of fixed lengths. Hamilton *et al.*

2017 recently introduced a "meta technique" known as "HARP," which allows graph preprocessing to enhance various random walking approaches [61]. In this approach, a coarsening procedure in the graph is used to collapse related nodes into "supernodes" in G, and then this coarsened graph runs DeepWalk, LINE or node2vec. After embedding the coarsened version of G, each supernode's learned embedding is used as its initial value for the random embedding of the constituents in the superstructure (a "fine-grained" version of the graph for a new round of nonconvex optimization). This cycle can be replicated hierarchically at varying coarseness rates and the output of DeepWalk, node2vec, and LINE has been consistently increased [61].

Dmitry [80] has provided an open access web-based platform tool called InfraNodus, which offers information from any text using data network analysis. The approach was used as a network and in a conversation based on the terms 'co-occurrence defines the most influent expressions. A network group discovery algorithm is then used to classify the various contextual clusters describing the key problems in the document and their relationships. In combination with other steps, the group composition is used to assess if the discourse is selective or cognitive complex. Furthermore, the conceptual holes in the graph will reflect the parts of the speech that lack links, thereby highlighting the places in which new concepts are possible. While standalone applications, the platform can be used both by end-users and implemented in other tools via an API.

B. GRAPH REPRESENTATION FOR WEB DOCUMENTS

Schenker *et al.*, 2005 proposed web document graph models (or general text documents), which included 6 graph methods for web documents: standard representation, simple representation, N-distance-representation, N-simple distance representation, absolute frequency representation, and relative frequency representation [11]. The adjacency of terms in an HTML file is the foundation of all these graph representations.

1) STANDARD REPRESENTATION

The first task under the standard representation is to identify terms that can be stemming or lemmas, etc., by using stemming algorithms or other language-specific standards, and each unique term in the document becomes a vertex in a graph that represents the document. Every vertex is labelled with the word it represents. In the text graph, the vertex labels are unique because for each word, a single vertex is generated even if a vertex appears in the text more than once. Second, if a word 'A' is immediately preceded by a word 'B' somehow in the \section ' (text information, title or reference, etc.) S of the text, then the representing vertex edge is the term 'A' to a vertex which is the term 'B' with the edge 'B', and a vertex is a vertex that corresponds to term 'A'. An edge is not linked between two terms if certain punctuation marks distinguish them. The graph will capture structural text information (relative term location, location) with this representation. For standard representation, there are three sections defined, such as title, text, and link. The title includes the text of the title of the document and all the keywords (metadata) given. Link is the anchor text that is shown in document hyperlinks. The text contains all text visible in the document (hyperlinked, not document titles and keywords). The text includes the content visible in the document. Graphs are language-independent representations, which means they can be applied in any language to a normalized document.

2) SIMPLE REPRESENTATION

The other form of a graph representation of [11] is referred to as the simple representation, which is fairly similar to the standard but the metadata or title are not examined, and the edges of this graph are not labelled.

3) N-DISTANCE REPRESENTATION

The third type of representation is defined as N-distance representation. This type only considers n-words and connects the successive words with an edge marked with the distance between the words (unless the terms are isolated by specific punctuation marks) rather than considering only words immediately following a certain word in the web document.

4) N-SIMPLE DISTANCE REPRESENTATION

An N-simple distance is a fourth graph representation type similar to N-distance in the graph representation idea. The difference is that the edges are not labelled, which implies the graph identifies only that the distance between two terms is n.

5) ABSOLUTE FREQUENCY REPRESENTATION

Absolute frequency representation resembles the simple representation type, but with additional frequency measurements. For vertices, it indicates how often the word has been included in the web document. The number of times between two connected words appears in the order defined for indicated edges.

6) RELATIVE FREQUENCY REPRESENTATION

The relative frequency is similar to the absolute frequency type in terms of graph representation. The normalized frequency parameters are related to the vertices and edges. The relative frequency representation considers the total number of word occurrences on the vertices and edges as well.

C. GRAPH-BASED REPRESENTATION IN NATURAL LANGUAGE PROCESSING

Some tasks of understanding natural language processing (NLP) depend on different types of structures of graphs, for example, word co-occurrence graphs, word-document graphs, sentences as graphs, and knowledge graphs.

The word co-occurrence graph can be identified as a local-context based word co-occurrence graph as well. In this type, words are assumed to occur with each other within a window. The main information is used by multiple models to learn

TABLE 2. Comparison between the graph text representation schemes.

Representation Type	Description Mechanism	Web Representation: Title & Meta-data	Hyperlinked	Text keywords	Example of Applications and Reference
Standard	Identify terms as vertex that can be stemming or lemmas Every vertex is labeled with the word it represents and immediately preceded by a word an edge will be created between them. The edges are labelled giving to title-*TI*, link-*L*, or text-*TX*.	Title includes the text of the title of the document and all the keywords (metadata) given Title is labelled	Link is the anchor text which is shown in document hyperlinks. Link is labelled	The text contains all text visible in the document (hyperlinked, not document titles and keywords). Text is labelled	Text Summarization[81], Text Classification[14, 29, 82] Plagiarism detection[44, 45, 47, 83, 84], Text categorization[85, 86] Sentiment classification[87]
Simple	Similar to the standard but the meta-data or title are not examined, and the edges of this graph are not labelled	meta-data or title are not examined, and the edges of this graph are not labelled	Link and hyperlinks are not labelled	Text are not labelled.	Text Summarization[88] DNA genomic sequences detection[89] Image captioning[90]
N-Distance	Only look to n words and connect the successive words with an edge marked with the distance between the words.	Title is labelled based on the distance(number of terms between two words)	Link is labelled	The text is labelled based on	Web Classification[29, 82] Pattern Recognition[91] [92-94]
N-Simple distance	Similar to the N-Distance. The different is that the edges are not labelled, which implies the graph identify only the distance between two terms is n.	Title is Not labelled based on the distance(number of terms between two words)	Link is Not labelled	The text is Not labelled based on	Document Clustering [95-97] Text categorization [13]
Absolute frequency	Rely on resembles of the simple representation type, but with additional frequency measurements. The number of times between two connected words appears in the order defined for edges are indicated.	The meta-data or title is labelled based on the frequency times associated with a term that appeared in the web document.	Link is labelled	The text is labelled based on	biomedical text summarization [98]
Relative frequency	Similar to the absolute frequency type. The normalized frequency parameters related with the vertices and edges. It is considers the total number of word occurrences on the vertices and edges as well.	The meta-data or title is labelled based on the frequency times associated with a term that appeared in the web document.	Link is labelled	The text is labelled based on	Assessing readability[99] Linguistic Representations[100]

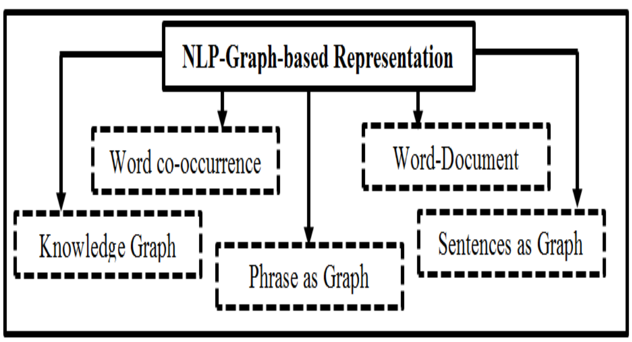


FIGURE 4. Taxonomy of NLP-graph-based representation.

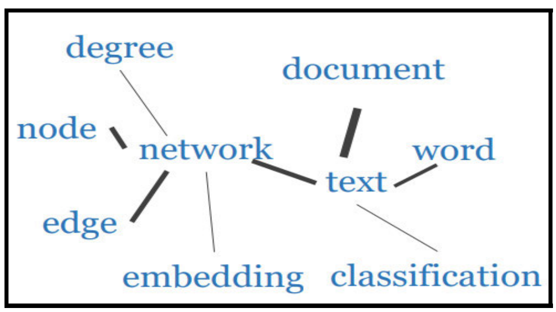


FIGURE 5. Word co-occurrence graph [101].

word embeddings e.g., SkipGram) [101] and global vectors for word representation (Glove) [102]. An example of the word co-occurrence graph is depicted in figure 5.

In the word-based document graph, information can be encoded about the occurrence of a word at the document level. The important information is used to study representations of words and documents. Models such as statistical topic models and paragraphs provide the main information e.g., latent Dirichlet allocation [103]. An example of the word-document graph is shown in figure 6.

The third type of the NLP graph-based representation is called sentences as graphs. In this type, the graph is represented as an encoding of the relationships of syntactic

TABLE 3. The analysis of the NLP graph-based representation types.

NLP Graph Representation	Description Mechanism	Graph Label Representation		Example of Research Applications Domain
		Nodes	Edges	
Word co-occurrence	Words are assumed to occur with each other within a window. The main information used by multiple models to learn word embedding.	Nodes are Labelled	Edges are Not Labelled	Keyword and Keyphrases Extraction[106-108] Biomedical domain[109] Machine translation[110]
Word-Document	Information can be encoded about the occurrence of word at the document level. The important information used to study representations of words and documents. Models such as statistical topic models and paragraph provide main information.	Nodes are Labelled	Edges are Not Labelled	Latent Dirichlet allocation[103]
Knowledge Graph co-occurrence	The graph in this type is represented by encoding the different entities ' relationships. For instance; Microsoft's Satori and Google's Freebase.	Nodes are Labelled	Edges are labelled	Question answering and Information search [104]
Phrase as Graph occurrence	The graph represented as encoded with a minimal automata a large set of phrases. The phrase-graph comprises of a node in any status update for each appearing phrase and an edge between each set of two phrases used adjacently in any status update.	Nodes are Labelled	Edges are Not Labelled	Plagiarism detection[44, 83, 84] Text Summarization[111] Text classification[29, 112] Text clustering[97, 113]
Sentences as Graph	The graph represented as encoding the relationships of syntactic and semantic dependence between words.	Nodes are Labelled	Edges are labelled	Machine translation[104], Semantic role labeling (SRL) Sentence classification[104] Social Media streaming [114] Text Summarization[115]

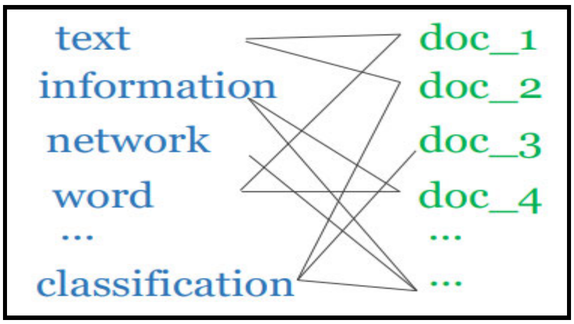


FIGURE 6. Word-document graph.

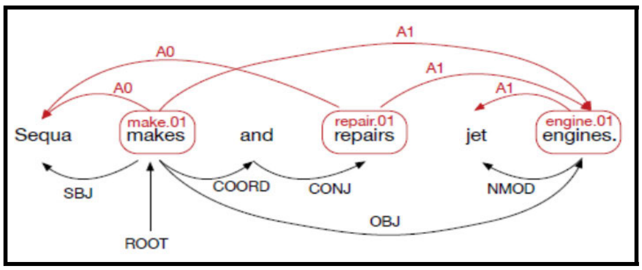


FIGURE 7. semantic and syntactic dependency graph [104].

and semantic dependency between words. This type is valuable for a diversity of tasks, such as machine translation and semantic role labelling (SRL) for sentence classification [104]. An example of the semantic and syntactic dependency graph is depicted in figure 7.

The fourth type is called a knowledge graph (KG). This type of graph is represented by encoding the different entities' relationships. Microsoft's Satori and Google's Freebase are examples of this type. The KG is suitable for question answering and information search tasks [105]. An example of the knowledge graph is shown in figure 8.

The sixth type represents the phrase of text as a graph (PG). The phrase of text is represented by two or more terms within the sentences. There is an overlap for identifying the phrase type between the word-based and sentence-based representation types. The concept behind phrase-graphs is generally simple: the graph is represented as an encoding with minimal automata of a large set of phrases. The phrase graph is composed of a node in any status update for each appearing phrase and an edge between each set of two phrases used adjacently in any status update. An example of the phrase graph is depicted in figure 8.

Table 3 shows the analysis of the NLP graph-based representation types. It focuses on the idea description for each type, graph label representation for the nodes and edges, and some of the research areas that implemented these types.

Table 3 offers flexible mechanisms to encode different structures of the graphs in natural language. Current progress on graph-based representation and learning provides an NLP understanding of opportunities for natural text and Internet

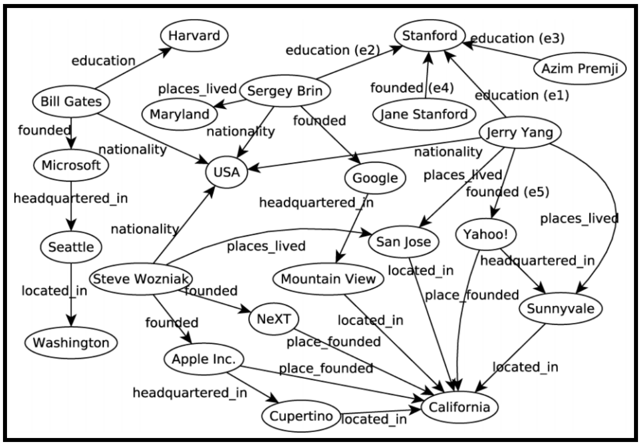


FIGURE 8. Knowledge graph (KG).

TABLE 4. Comparison between the NLP graph text representation schemes and its limitation.

Graph Representation Method	Attributes and Parameters	Limitations
Co-occurrence Type[3-5, 116]	Terms/Sentence-closeness	window size of Co-occurrence
Collocation-based Relation[117]	Terms/sentence-closeness	Limit of basic assumptions
POS Tagger-based[118]	Tagger with mapping	External POS-Tagger not supported
Parse-thicket[119]	Parser	Lacking of Grammatical relations
Semantic-based Relation[10, 120, 121]	Context-based	External ontologies not supported
Concept-based[122-124]	Relative among concepts	External ontologies not supported
Hierarchical-keyword [125]	Term-closeness Relative among concepts	External ontologies not supported, Limit of window size

webpages. We noted that each term is represented with word co-occurrence graphs, and each document and sentence is represented with heterogeneous text and a sentence graph, respectively.

Another investigation and analysis of graph-based representation based on the graph method attributes and limitations was conducted by [126] in table 4. It presents a detailed overview of methods that reflect the text document as a graph. It focused on the two components of parameters and limitations. The parameters are a key component taken into account during the construction of a graph. However, the limitations are disadvantages of the techniques that the specified method extremely relies on given the listed parameters.

D. GRAPH MATCHING

Graph matching, which involves a group of computational problems to find the best match between the vertices of the graphs by minimizing (maximizing) node and edge discrepancies, is a key issue in computer science and covers numerous areas, including combinatorics, pattern recognition, multimedia, and computer vision. Inexact weighted-graph matching receives more attention because of its flexibility and practical utility compared with the exact graph (sub) isomorphism frequently considered in a theoretical setting. One of the main advantages of the relation information is that graphs allow a stronger representation of structural relations through a graph rather than a vector. The nodes and borders with arbitrary attributes are generally assigned. There are two general categories of the graph matching problem: the exact match and the inaccurate matching. A strict correspondence or one that exists at least between their substructures must be found in the previous mission. In the latter case, this requirement is relaxed to find the opposition between the nodes that optimizes a certain criterion of affinity or distortion; thus, it is also referred to in the literature as tolerance to error/correct graph matching [127], and for real-world issues, the matching of non-identical graphs has to be dealt with. The matching phase involves the inspection of candidates that were determined during the candidate selection process where they were tested against a specified pattern. Various matching algorithms have been proposed and may be classified as either search-based (optimal methods) or numerical-based (approximate methods) [128]. In determining the similarity of two graphs, the calculation is far more complicated compared to calculating the similarity of two vectors. This is due to the graph containing shape information, and as such, serious time efficiency concerns are prevalent during computation. Recently, some graph similarity metrics, including a distance measure based on the common maximum subgraphs and subgraph detection algorithms have materialized. Wallis *et al.* [94] and Bunke and Shearer [92] used a combination of a maximum common subgraph and a minimum common supergraph as a graph similarity measure. For the calculation of similarities among objects described by attributed connected graphs, a new graph distance metric is suggested [93]. The algorithm that performs an error-correction graph matching while running in accordance with an appropriate cost function can calculate the proposed metric, and the extension only takes linear time with respect to the size of the graphs. Gao and Gao [129] proposed an optimal approach to calculating graph similarities. Through adding connected subgraphs in the kernel graph, they obtained a low-dimensional structure vector. Subgraphs were then compared and the comparability of the respective subgraphs was measured. The study used some examples to demonstrate the viability of the suggested approach.

Traditional methods for calculating the maximum common subgraph between two text graphs are generally derived from the maximum group finding or back-tracking methods. Theoretically, these methods achieved a high time efficiency as exemplified by the worst-case time efficiency of the algorithm, which is equivalent to, where m and n represent the number of vertexes within the graphs that were considered. Relevant studies regarding pattern matching in graphs have

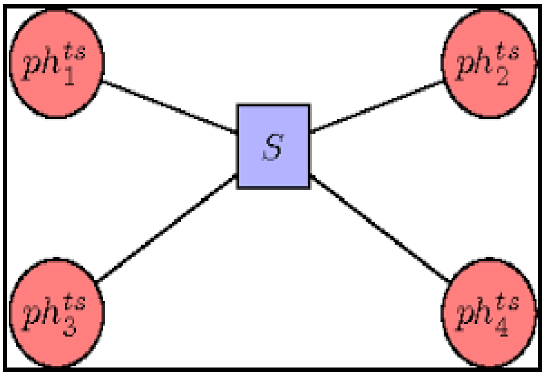


FIGURE 9. Phrase graph (PG).

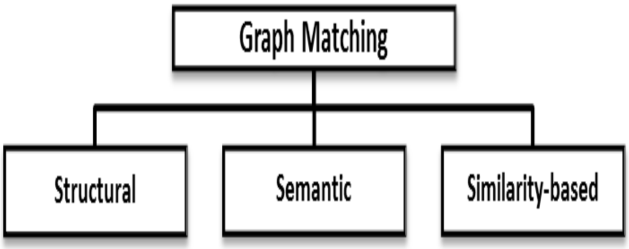


FIGURE 10. Main types of graph matching approaches.

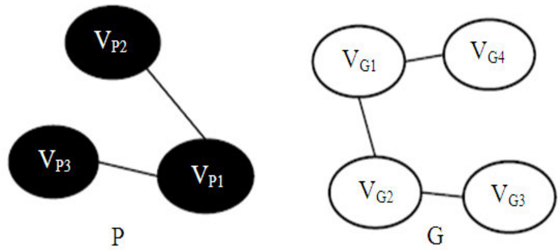


FIGURE 11. An example pattern graph P and data graph G [130], [131].

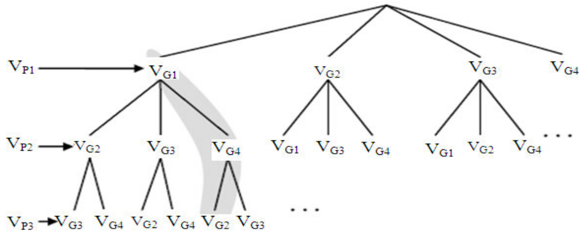


FIGURE 12. A partial search tree for Ullmann's algorithm, mapping vertices from pattern graph P to data graph G [130], [131].

been conducted by various research communities within and beyond computer science and [130]. Areas of application and pertinent research fields include information retrieval, databases, mathematical graph theory, computer vision, artificial intelligence, computer-aided design, biology, electronics, data mining, and knowledge discovery. Graph-based pattern matching is a set of related problems as opposed to merely being a single problem [131]. These issues include the whole NP subgraph isomorphism issue, which relies heavily on the graphic structure and is not accurately matching complex patterns with thousands of typed and attributed vertices and edges in semantic graphs. In graph structure and semantics, specific approaches for accurate and inaccurate matching are set. Descriptive, but non-comprehensive, approaches are provided here. There are different types of graph matching approaches, as shown in figure 10.

1) STRUCTURAL MATCHING APPROACH

Ullmann, 1976 proposed a structural matching approach that included a subgraph isomorphism algorithm [132]. Ullmann's method was one of the earliest approaches of exact pattern matching and was used on single untyped graphs that had either undirected or directed edges. Figure 11 illustrates how matches to the pattern graph P in the data graph G were found in Ullmann's method. At its core, this algorithm worked by using a depth-first tree search algorithm to specify all the potential mappings of the vertices in G to the vertices in P. Figure 12 shows how at level i of the search tree, each node maps vertex VPi in P to a vertex in G [130] and [131].

The highlighted path represented a match for P in G [130], [131]. In the above figures, the vertices in P are mapped vertices in G. If the adjacency between P and G is retained, then those vertices are said to be neighbours. As a result, an isomorphism from P to a subgraph of G is represented. On the other hand, if there is no adjacency to maintain between P and G, then P and G are not neighbours, and consequently, no isomorphism is present. Ullmann went on to recommend that the process be refined and the search tree be pruned to remove subtrees. As a result of pruning, the search space used by this method was reduced. The process outlined did not consider vertex mappings. These were omitted using the following three criteria:

- ***Vertex Degree:*** *The first criteria for omitting vertex mapping stated if the degree of vertex VPi is greater than the degree of VGj, then VPi cannot map to VGj.*
- ***One-to-one mapping of vertices***: *To map VPi-VGj, along a certain path through the tree, it is not possible to map VPi to any other vertex in G, nor can any other vertex in P map to VGj.*
- ***Forward checking***: *The next step is to eliminate all remaining possible vertex mapping if it does not remain a neighbour to either P or G. In the above example, the mapping from VP2-VG3 is omitted.*

There are two situations that can result from creating a specific path in the search tree using Ullmann's algorithm. In the first situation, the algorithm may omit all the possible mappings from some of the vertexes in P. Consequently, the path will not be capable of providing a match. This process can be stopped without consequence in regard to the additional nodes along the path. In the second scenario, the algorithm maps all the way to a leaf in the tree, and each vertex in P is mapped to a vertex in G. The resulting path corresponds to a match for P in G (Figure. 8). As observed by Messmer and Bunke [133], Ullmann's algorithm has exponential worst-case time-complexity regardless of the refinement process. As a result, they developed an alternate way to extract subgraph isomorphism. In this technique, the graph dataset is pre-processed. This allows the likely changes in the graph adjacency matrix to be used to build the decision tree. The decision tree will categorize the adjacency matrix of the pattern graph. Pruning techniques, as suggested by Messmer and Bunke [124] should be applied at this time to reduce the size of the decision tree so that any benefits are not negated by the tree's exponential growth. McKay, 1990 used the Nauty algorithm to detect isomorphism among untyped graphs that may be directed or undirected [134]. The Nauty algorithm reduced graphs into a conical form. This allowed for the speedy discovery of isomorphism [135]. The Nauty algorithm then computed the invariants for each graph vertex. As a result, the graph was divided into a non-overlapping set of vertices. These vertices were based on invariant values. Next, any set containing the same invariant values were compared between graphs. A graph was said to be isomorphic if all the sets between the two graphs were isomorphic. Consequently, the requirement of testing for isomorphism between sets if the two graphs contained sets with different invariants became obsolete. Cook and Holder, 1995 developed a system called SUBDUE [136]. SUBDUE operated in a single graph setting containing typed and typed directed edges. Under SUBDUE, a path through a decision tree relates to a complete map or vertices. The matching capability of SUBDUE is inaccurate and as a result, each node in the search tree contains a value that sets out the degree of similarity between P and G. For example, if P and G are exactly isomorphic, they would be assigned a value of 0. These values rely on the graph edit distance [137]. The graph edit distance measures the minimum number of edit operations (deletions, insertions, and substitutions of edges and vertices) needed to change one graph into another graph. A branch and bound search was another feature of SUBDUE. This search was applied to solve the problem of the large search space. This algorithm also allowed for considerable time savings because it permitted a limit to be placed on the number of search nodes that would be searched. Unfortunately, the savings in time came at the expense of quality and the end solutions were not as good as they could have been.

2) SEMANTIC MATCHING APPROACH

A semantic graph is the graph-based display of information, where the vertices represent concepts (e.g., film, actor) and the edge of which is connected (e.g., appearance). Both vertices and edges are typed and assigned in a semantic graph. In addition, a semantic graph has the associated ontology, which defines the possible concepts, the possible relations between each concept pair and the attributes linked to each concept and relation.

To date, there have been several methods used to match texts based on the concepts of the texts. Early techniques were pioneered by [138] and [139]. Both teams relied upon a combination of graph structure and individual graph element attributes to uncover the common elements between graphs. Both teams also employed search algorithms and pruning techniques. The technique introduced by [138] employed an exact structural match. They suggested that any calculations done to determine the probability of attribute differences should be based on the results of the data. In cases where there is no data, Tsai and Fu recommended using the weighted distance, and weighted square error distance measure could be used instead. On the other hand, Shapiro and Haralick,1981 proposed a method that defined graphs as ''matching'' if the number of differences between structures of the graphs was within a predetermined limit [140]. A higher value was placed on the more important structural elements, and these elements influenced how closely one graph could be said to match another graph. The graph edit distance used by SUBDUE to determine the level of similarity between graphs can also be used to determine the level of semantic similarity since the values determined by the edit function can be used for semantic elements rather than purely structural elements. In this instance, the possible variations in the values of vertex substitutions are examined [136]. The kind of information discovered by the vertex was inherent to the method proposed by [141]. Their method used vertex type information in their algorithm in the graph-transaction setting on undirected graphs with typed vertices; matching was determined by the idea of a ''label path'', otherwise defined as the series of label types found along a specific path in a graph. A fingerprint for each was created during the construction of the dataset index using an algorithm. The fingerprint for a graph resulted in a pair set, where one component referred to the label path (h), which is a hash function, and the other component referred to the unique label path (count), which is related to the number of times the unique path occurred in a graph.

OntoSeek is another technique that attempted to match semantic similarities between documents [142]. OntoSeek matched documents by defining each document as a conceptual map. OntoSeek then measured the degree of semantic similarity between graphs. This process could only be carried out if there was an exact structural match between the query and a subgraph of a corresponding document. However, matches could only be discovered if the concept put forth in the query is a generalization of the concept expressed in the document. Matches are found by first checking the least probable links so that non-matches can be discarded from further consideration. The TMODS system as discussed by [142] and [143] considered directed attribute graphs. In this system, genetic algorithms were used to find exact and inexact pattern matches. TMODS focused on patterns because it was assumed that the patterns express both structural and attribute characteristics. TMODS searched patterns from the bottom to the top after the sub-patterns are identified. Once the sub-patterns have been determined, more complex, higher-level patterns can be examined. TRAKS is yet another algorithm used in past attempts at semantic matching, as discussed by [144]. TRAKS performs inexact pattern matching in typed, directed graphs. The ontological distance between types was ranked according to how close the type matched the original pattern. To decrease the time needed to run TRAKS, the pattern's components were processed in ascending order

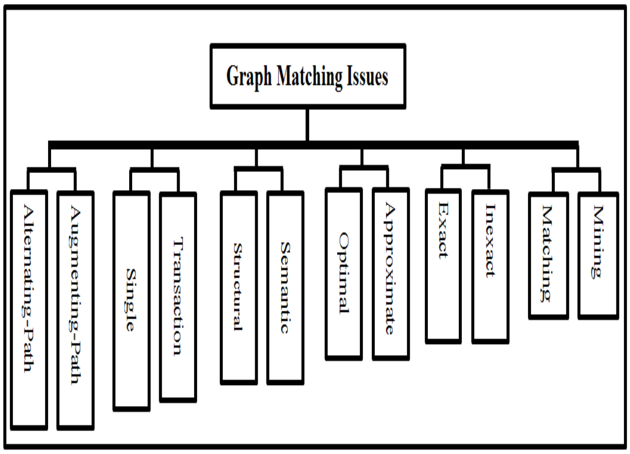


FIGURE 13. Main issues of graph matching approach.

according to how often their type appeared. This step allowed for quick identification and elimination of non-matches.

3) SIMILARITY-BASED MATCHING APPROACH

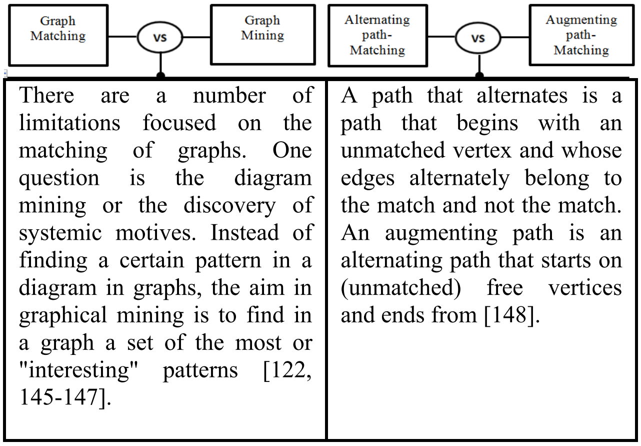
The inexact matching approaches examined in the preceding section all relied on the similarity between graphs as a way of matching semantic elements. The criteria for selecting a match depended on a similarity measure. The similarity measures consider the possible type, attribute and structural information of each distance. Some of the approaches also used a graph edit distance that was discussed above. Graph edit distances can identify semantic similarities but there are drawbacks. Each edit operation requires that a description of each value be provided. It is uncertain whether any benefit can be found in regard to the resulting distance measure if time is taken to allocate a description to each value. In light of the drawbacks in using graph edit distances to identify semantic similarities, Bunke and Shearer, 1998 suggested using distance metrics derived from the maximal common subgraph of [92] and minimum common supergraph of [145] as a solution to the graph edit distance problem. Burke's metrics measured the structural overlap between graphs of [145]. As a result, the constraints on an edited value are displayed.

A simple equation can be used to compute both metrics associated with the graph edit distance. Additionally, the attribute values can be compared to any similarity measures, including data type-reliant similarity values such as Euclidean distance or more general measures. Attempts have been made to formalize a theory that captures the complexity behind similarity-based graph matching. Bunke,1999 proposed a definition of error-correcting graph matching where the edit-based matching does not rely on the values given to individual edit operations [127]. Instead, Bunke suggested that edit-based matching should rely on the ratio of the values given to individual edit operations. Additionally, Berry and Sigayret, 2004 have shown that the root of graph similarity measures can be found in the theory of inexact pattern matching that views patterns only as a way of ranking possible matches [146]. The comparison between the graph matching issues and approaches is shown in figure 13.

4) SUMMARY AND ANALYSIS BETWEEN THE MAIN GRAPH MATCHING APPROACHES AND ITS LIMITATIONS

Structural Matching VS Semantic Matching	Single-Graph VS Transaction Graph
Graphs can be designed to match graphs based not strictly on their similarity as graphs but on their different representations in certain fields of interest, such that graphs may serve as conceptual representation. Since the importance of semantic graphs is primarily found within the type and attribute of the information stored on individual vertices and edges, structural correspondence is often not enough for similar graphs [132, 134, 135]. Semantic similarity tries to match graphs by taking into consideration vertex, edge forms, attributes and graph form, based on their context [143, 144, 147]	Data from graphs can be a single large diagram or rather a small set of graphs (often referred to as transactions). Both instances are called a single-graph configuration or a node transaction configuration[148]. The single-graph design is more general, and algorithms can be easily applied to a graph transaction, although the same is not true in general. During this study, we use the generic term graph dataset (or just dataset) for a collection of graph-structure data, regardless of whether it is structured as a single large diagram or as many small diagrams.

Exact Matching VS Inexact Matching	Optimal solutions VS Approximate solutions
A graph matched method can only return results that exactly suit a given pattern, or a list of the most similar matches is returned (i.e., inaccurate matches). Inexact matching algorithms are often called error-correction algorithms because they allow matching in the event of noise or data errors[129, 140]. Moreover, other systems only allow partially defined patterns (e.g., use wild cards or cardinal operators) to be left. If these results exactly match with the pattern, then the pattern itself is insufficient and considered as inexact matching (for example, "find all film vertices," compared with "find all film vertices connected with the vertex of an actor")[141, 142].	Because of whether an algorithm matches as inexact or exact, algorithms differ in performance guarantees in terms of solution. Optimum algorithms ensure the correct solution (e.g., a set of exact subgraphs matching the pattern; the nearest match or correctly defined match-list for an incorrect match) but exhibit exponential complexity [130, 143, 144] and are often of polynomial complexity but are not guaranteed to find the appropriate solution (for example, most, but not all matches for exact matching; for an inaccurate match, a close match, but not the closest ones). Optimal algorithms tend to be numerical and search-based in general [120].



Some studies has been provided the time complexity analysis for graph matching algorithms such as Sun *et al.* [149]. Table 5 shows the analysis of the subgraph matching time complexity.

Table 5 demonstrates the costs index of a some illustrative subgraph similarity studies proposed by Cordella *et al.* [150], Ullmann [132], Neumann and Weikum [151],Holder *et al.* [152], Atre *et al.* [153], Zhu *et al.* [154], Zou *et al.* [155], Cheng *et al.* [156], He and Singh [157], Zhang *et al.* [158], Zhao and Han [159] (2010), and Sun *et al.* [149].

Table 6 represents the overview of several technologies including semantic matching, such as schema creation, event analysis, information integration, knowledge diversity management, query translation, and resource discovery, graph matching tools, and algorithms that have been proposed.

IV. OPEN PROBLEMS AND RESEARCH GAPS

Over the last few decades, it was an open challenge to develop algorithms that were ideal for large-scale graphs of low complexity. In the area of text representation and graphics learning, a number of practical open problems remain to be addressed.

1. While most of the studies we reviewed are extremely scalable in graph theory (i.e., V(|E|) representation), there is still an important study to be done in scaling vertex and graph representation methods to truly massive text documents (e.g., billions of vertices and edges). For instance, most approaches rely on representing and storing a unique graph for each individual text. Furthermore, the assessment setups adopt that the lists of vertices and edges of all graphs used for text representation can fit in computer memory, a supposition that is at dispute with the reality of many applications domains, wherever graphs are evolving, massive, and sometimes kept in a spread style. To avoid the widening of the disconnections between the academic research community and the user implementation of these methods, the design of a text representation system that is fully applicable to practical production environments is required.
2. Although there are many studies that have represented texts in the form of graphs and used them to solve their problem issues, these methods semantically lack the representation of the textual meanings in terms of knowing the linguistic concept of texts and then addressing them in their research issues. In this aspect, certain methods do not consider individual words and instead take the whole sentence as one unit for graph representation. However, the semantic similarity between the represented text and graph is not captured if the users modified some sentences using paraphrasing or word replacement.
3. The quantification of semantic matching in the language is the core of many applications for NLP and AI. Specific types of linguistic objects such as single word meanings or full sentences are usually limited to those steps. Therefore, several measurements of semantic matching, which often use different internal representations or have different output scales, are required for an application downstream to accommodate different types of data.
4. There are vast challenges in determining the appropriate software that is used to represent texts as a graph. This determination requires a great effort in the process of representing texts as a graph; additionally, preserving the real content of the text after representation is required.
5. While there are many techniques that used the similarity of graphs and graph matching, a computational time problem still exists. The process of comparing between two graphs takes a long processing time for nodes and edges between the graphs because the representation of the text as a graph may generate a huge number of nodes and edges per graph; thus, the matching time becomes very large. We need a convincing and accurate method of graph similarity to produce accurate matching results with less computational time.
6. In the subgraph, a major technical drawback for current subgraphs is that before the learning process, the target subgraphs have to be initialized. However, several methods aim to find subgraphs with certain properties,

TABLE 5. Time complexity comparison between the subgraph algorithms.

Algorithm	Index-size	Index-time	Update-cost	Experimental Graph-size
Ullmann & VF2	-	-	-	4484
RDF-3X	O(m)	O(m)	O(d)	33M
BitMat	O(m)	O(m)	O(m)	361M
Subdue	-	-	Exponential	10K
SpiderMine	-	-	Exponential	40K
R-join	$O(nm^{1/2})$	$O(n^4)$	O(n)	1M
Distance-join	$O(nm^{1/2})$	$O(n^4)$	O(n)	387K
GraphQL	O(m + ndr)	O(m + ndr)	O(ndr)	320K
Zhao	O(ndr)	O(ndr)	O(ndL)	2M
GADDI	O(ndL)	O(ndL)	O(ndL)	10K
STwig	O(n)	O(n)	O(1)	1B

TABLE 6. Summary of the graph tools schemes and its characteristics.

Year	Method and Tools	Index Structure	Matching structure	Location (In/ Out Memory) / on cloud	Application Domain	Ref
1976	Ullman	No-Index	Exact-Matching	In-M	subgraph isomorphism	[132]
1994	SubDue	Frequent-subgraph	Inexact-Matching	Out-M	Knowledge discovery from structural data, graph matching	[136]
2002	GraphGrep	Frequent-subgraph	Exact & Inexact-Matching	In-M	querying graphs	[141]
2004	Gindex	Inverted-Index	Exact & Inexact-Matching	Out-M	isomorphism search and Graph matching	[160]
2006	CTree	Frequent-subgraph	Exact & Inexact-Matching	In-M	Data Regression and classification	[161]
2007	FG-Index	Edge-Index	Exact-Matching	Out-M	subgraph isomorphism querying graphs	[162]
2007	Tree+Δ	Tree-based Index	Exact-Matching	In-M	querying graphs	[163]
2008	TALE	NH-Index	Inexact-Matching	Out-M	graph-based dependency parser	[164]
2008	GraphQL	Frequent-subgraph	Exact & Inexact-Matching	In-M	Subgraph isomorphism based pruning	[157]
2008	RJOIN	Frequent-subgraph	Exact & Inexact-Matching	In-M	Subgraph isomorphism search	[156]
2008	QuickSi	Swift-Index	Exact-Matching	In-M	isomorphism search and Graph matching	[165]
2009	GADDI	NH-Index	Exact-Matching	In-M	isomorphism search and Graph matching	[158]
2009	DistanceJoin	Frequent-subgraph	Inexact-Matching	In-M	isomorphism search and Graph matching	[155]
2010	BitMat	Edge-Index	Exact-Matching	In-M	subgraph isomorphism querying graphs	[153]
2010	Rdf-3x	Edge-Index	Inexact-Matching	Out-M	subgraph isomorphism querying graphs	[151]
2010	SIGMA	Frequent-subgraph	Inexact-Matching	On Cloud	Graph matching based on query path	[166]
2010	Spath	Edge-Index	Inexact-Matching	In-M	Graph matching based on query path	[159]
2011	SpiderMine	Frequent-subgraph	Exact-Matching	Out-M	subgraph isomorphism querying graphs	[154]
2011	Ness	Vertex-Index	Exact & Inexact-Matching	Out-M	Subgraph isomorphism search and graph similarity	[167]
2012	STwig	No-Index	Exact & Inexact-Matching	On Cloud	Subgraph isomorphism search	[149]
2013	Turbo*ISO*	No-Index	Exact-Matching	In-M	Subgraph isomorphism search	[168]
2014	Duall*ISO*	Vertex-Index	Exact-Matching	In-M	Subgraph isomorphism search	[169]
2015	VF2Plus(VF 2++)	Vertex-Index	Exact-Matching	In-M	bioinformatics applications	[170]
2016	Sum*ISO*	Vertex-Index	Exact-Matching	In-M	Subgraph isomorphism search	[171]
2017	BB-Graph	Vertex-Index	Exact-Matching	In-M	isomorphism search and Graph matching	[172]
2017	VF3	Vertex-Index	Exact-Matching	In-M	bioinformatics applications	[173]
2018	PLGCoding	Vertex-Index	Exact-Matching	In-M	graph querying	[174]
2019	InfMatch	Vertex-Index	Exact & Inexact-Matching	In-M	Graph Matching	[175]

and such implementations require models that can be focused on the combination of a wide range of subgraphs.

7. In the graph embedding approach, learning representation is desirable because it relieves much of the stress of hand-designed characteristics, but it also has a well-known interpretability price. We believe that embedding methods have efficient algorithms, but these algorithms remain relatively unknown regarding fundamental limitations and potential underlying

biases. To proceed, new techniques must be developed to improve the interpretability of the knowledge, beyond visualization and benchmarking. In light of the complexity and capacities of these techniques, scientists must always be careful to ensure that they are truly able to represent their methods.

V. CONCLUSIONS AND FUTURE WORK

In this review, we conduct an inclusive and broad survey of the state of the art in graph-based text representation. The survey provides basic definitions of the structure of graph-based text representations and proposes a new taxonomy for the main issues related to graph-based text representation. A sub-taxonomy of graph models for web documents has been introduced and categorized into six main types based on their functionality, which include standard representation, simple representation, N-distance-representation, N-simple distance representation, absolute frequency representation, and relative frequency representation. More significantly, the paper provides two taxonomies of the NLP-based graph and graph matching taxonomy to classify the current studies in graph structure and graph matching methods, respectively. For the NLP-based graph taxonomy, we describe five categories of NLP-graph representation with their mechanisms and conclude the limitations faced in each category. On the other hand, the graph matching taxonomy discusses three main types, including structure-, semantic-, and similarity-based matching. The analysis between the graph matching issues and approaches has been summarized and reported by highlighting their challenges. In addition, the development of the graph matching tools and methods over the past years has been presented and reported in terms of the concept of matching, locality, indexing feature and structure, and the application domain that employed these tools. Finally, we recommend seven promising future study directions in the graph-based text representation field. The open problems and challenges of graph-based text representation and learning are elaborated in order to exploit the limitations and research gaps to guide scientific researchers in identifying adequate solutions.

As future work, we will expand this survey with other graph representation phases and fields and link it with other related fields. In addition, we will propose and suggest potential solutions to the discussed problems to fill the summarized research gap.

ACKNOWLEDGMENTS

This work was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant no. DF-681-830-1441. The authors, therefore, acknowledge with thanks DSR technical and financial support.

REFERENCES

[1] G. Hirst, "Semantic interpretation and ambiguity," *Artif. Intell.*, vol. 34, no. 2, pp. 131–177, Mar. 1988.
[2] Y. Wang, X. Ni, J.-T. Sun, Y. Tong, and Z. Chen, "Representing document as dependency graph for document clustering," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 2177–2180.
[3] W. Jin and R. K. Srihari, "Graph-based text representation and knowledge discovery," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2007, pp. 807–811.
[4] F. Zhou, F. Zhang, and B. Yang, "Graph-based text representation model and its realization," in *Proc. 6th Int. Conf. Natural Lang. Process. Knowl. Eng. (NLPKE-)*, Aug. 2010, pp. 1–8.
[5] F. Rousseau and M. Vazirgiannis, "Graph-of-word and TW-IDF: New approach to ad hoc IR," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 59–68.
[6] V. KumarSihag and S. Kumar, "Graph based text document clustering by detecting initial centroids for k-Means," *Int. J. Comput. Appl.*, vol. 62, no. 19, pp. 1–4, 2013.
[7] M. S. Hossain and R. A. Angryk, "GDClust: A graph-based document clustering technique," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Oct. 2007, pp. 417–422.
[8] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 1, pp. 39–41, 1995.
[9] J. Violos, K. Tserpes, E. Psomakelis, and K. Psychas, "Sentiment analysis using word-graphs," in *Proc. 6th Int. Conf. Web Intell., Mining Semantics (WIMS)*, 2016, p. 22.
[10] J.-Y. Chang and I.-M. Kim, "Analysis and evaluation of current graph-based text mining researches," *Adv. Sci. Technol. Lett.*, vol. 42, pp. 100–103, Dec. 2013.
[11] S. Adam and B. Horst, *Graph-Theoretic Techniques for Web Content Mining*, vol. 62. Singapore: World Scientific, 2005.
[12] A. Markov, M. Last, and A. Kandel, "The hybrid representation model for Web document classification," *Int. J. Intell. Syst.*, vol. 23, no. 6, pp. 654–679, Jun. 2008.
[13] A. Markov, M. Last, and A. Kandel, "Fast categorization of Web documents represented by graphs," in *Proc. Int. Workshop Knowl. Discovery Web*, 2006, pp. 56–71.
[14] C. Jiang, F. Coenen, R. Sanderson, and M. Zito, "Text classification using graph mining-based feature extraction," in *Research and Development in Intelligent Systems XXVI*. London, U.K.: Springer, 2010, pp. 21–34.
[15] T. N. Q. Do and A. Napoli, "A graph model for text analysis and text mining," M.S. thesis, Université de Lorraine, Grand Est, France, 2012.
[16] A. Formica, "Similarity reasoning in formal concept analysis: From one-to many-valued contexts," *Knowl. Inf. Syst.*, vol. 60, no. 2, pp. 715–739, Aug. 2019.
[17] C. A. Kumar and P. K. Singh, "Knowledge representation using formal concept analysis: A study on concept generation," in *Global Trends in Intelligent Computing Research and Development*. Hershey, PA, USA: IGI Global, 2014, pp. 306–336.
[18] B. Ganter, S. Rudolph, and G. Stumme, "Explaining data with formal concept analysis," in *Reasoning Web. Explainable Artificial Intelligence*. Bolzano, Italy: Springer, 2019, pp. 153–195.
[19] G. Stumme, "Formal concept analysis on its way from mathematics to computer science," in *Proc. Int. Conf. Conceptual Struct.*, 2002, pp. 2–19.
[20] A. K. Sarmah, S. M. Hazarika, and S. K. Sinha, "Formal concept analysis: Current trends and directions," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 47–86, Jun. 2015.
[21] P. Cimiano, A. Hotho, and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis," *J. Artif. Intell. Res.*, vol. 24, pp. 305–339, Aug. 2005.
[22] U. Priss, "Formal concept analysis in information science," *Annu. Rev. Inf. Sci. Technol.*, vol. 40, no. 1, pp. 521–543, 2006.
[23] L. Wang and X. Liu, "A new model of evaluating concept similarity," *Knowl.-Based Syst.*, vol. 21, no. 8, pp. 842–846, Dec. 2008.
[24] A. El Qadi, D. Aboutajedine, and Y. Ennouary, "Formal concept analysis for information retrieval," 2010, *arXiv:1003.1494*. [Online]. Available: http://arxiv.org/abs/1003.1494
[25] K. Rajaraman and A.-H. Tan, "Knowledge discovery from texts: A concept frame graph approach," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 669–671.
[26] K. Rajaraman and A.-H. Tan, "Mining semantic networks for knowledge discovery," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 633–636.
[27] J. F. Sowa and E. C. Way, "Implementing a semantic interpreter using conceptual graphs," *IBM J. Res. Develop.*, vol. 30, no. 1, pp. 57–69, Jan. 1986.
[28] M. Montes-y-Gómez, A. López-López, and A. Gelbukh, "Information retrieval with conceptual graph matching," in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2000, pp. 312–321.

[29] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of Web documents using a graph model," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 240–244.
[30] S. Chu and B. Cesnik, "Knowledge representation and retrieval using conceptual graphs and free text document self-organisation techniques," *Int. J. Med. Informat.*, vol. 62, nos. 2–3, pp. 121–133, Jul. 2001.
[31] P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, and N. Maeda, "The transcriptional landscape of the mammalian genome," *Science*, vol. 309, no. 5740, pp. 1559–1563, Sep. 2005.
[32] S. Hensman and J. Dunnion, "Constructing conceptual graphs using linguistic resources," in *Proc. 4th Int. Conf. Telecommun. Inform. (WSEAS)*, in World Scientific and Engineering Academy and Society (WSEAS), M. Husak and N. Mastorakis, Eds., Stevens Point, WI, USA: World Scientific, 2005, pp. 1–6.
[33] A. K. Abdulsahib, "Graph based text representation for document clustering," M.S. thesis, Univ. Utara Malaysia, Changlun, Malaysia, 2015.
[34] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *Proc. 6th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2013, pp. 465–474.
[35] S. Ordoñez-Salinas and A. Gelbukh, "Information retrieval with a simplified conceptual graph-like representation," in *Proc. Mexican Int. Conf. Artif. Intell.*, 2010, pp. 92–104.
[36] F. Balmas, "Displaying dependence graphs: A hierarchical approach," *J. Softw. Maintenance Evol., Res. Pract.*, vol. 16, no. 3, pp. 151–185, May 2004.
[37] T. Zimmermann and N. Nagappan, "Predicting defects using network analysis on dependency graphs," in *Proc. ACM/IEEE 30th Int. Conf. Softw. Eng.*, 2008, pp. 531–540.
[38] J. Dietrich, V. Yakovlev, C. McCartin, G. Jenson, and M. Duchrow, "Cluster analysis of java dependency graphs," in *Proc. 4th ACM Symp. Softw. Vis. SoftVis*, 2008, pp. 91–94.
[39] C. Patel, A. Hamou-Lhadj, and J. Rilling, "Software clustering using dynamic analysis and static dependencies," in *Proc. 13th Eur. Conf. Softw. Maintenance Reeng.*, 2009, pp. 27–36.
[40] S. Chakravarthy, A. Venkatachalam, and A. Telang, "A graph-based approach for multi-folder email classification," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 78–87.
[41] B. S. Mitchell and S. Mancoridis, "Clustering module dependency graphs of software systems using the bunch tool," Nat. Sci. Found., Alexandria, VA, USA, Tech. Rep., 1998.
[42] F. Beck and S. Diehl, "On the impact of software evolution on software clustering," *Empirical Softw. Eng.*, vol. 18, no. 5, pp. 970–1004, Oct. 2013.
[43] I. Valatkaite and O. Vasilecas, "Automatic enforcement of business rules as ADBMS triggers from Conceptual Graphs model," *Inf. Technol. Control*, vol. 31, pp. 36–42, 2004.
[44] A. H. Osman, N. Salim, M. S. Binwahlan, R. Alteeb, and A. Abuobieda, "An improved plagiarism detection scheme based on semantic role labeling," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1493–1502, May 2012.
[45] A. H. Osman, N. Salim, M. S. Binwahlan, H. Hentabli, and A. M. Ali, "Conceptual similarity and graph-based method for plagiarism detection," *J. Theor. Appl. Inf. Technol.*, vol. 32, pp. 135–145, 2011.
[46] A. H. Osman, N. Salim, Y. J. Kumar, and A. Abuobieda, "Fuzzy semantic plagiarism detection," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.*, 2012, pp. 543–553.
[47] A. Hamza Osman, N. Salim, and M. Salem Binwahlan, "Plagiarism detection using graph-based representation," 2010, *arXiv:1004.4449*. [Online]. Available: http://arxiv.org/abs/1004.4449
[48] K. Toutanova, A. Haghighi, and C. D. Manning, "Joint learning improves semantic role labeling," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 589–596.
[49] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, Apr. 1958.
[50] P. B. Baxendale, "Machine-made index for technical literature—An experiment," *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 354–361, Oct. 1958.
[51] E. Hovy and C. Y. Lin, "Automated text summarization in SUMMARIST," in *Advances in Automatic Text Summarization*, vol. 14. Cambridge, MA, USA: MIT Press, 1999, pp. 81–94.
[52] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969.
[53] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Adv. Autom. Text Summarization*, vol. 3610, pp. 111–121, 1999.
[54] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Comput. Linguistics*, vol. 17, no. 1, pp. 21–48, Mar. 1991.
[55] A. M. El-Said, A. I. Eldesoky, and H. A. Arafat, "Exploiting semantic annotations and-learning for constructing an efficient hierarchy/graph texts organization," *Sci. World J.*, vol. 2015, Jan. 2015, Art. no. 136172, doi: 10.1155/2015/136172.
[56] K. Yang and L. Toni, "Graph-based recommendation system," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 798–802.
[57] H. Peng *et al.*, "Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8933476, doi: 10.1109/TKDE.2019.2959991.
[58] Z. Huang, W. Chung, T.-H. Ong, and H. Chen, "A graph-based recommender system for digital library," in *Proc. 2nd ACM/IEEE-CS Joint Conf. Digit. Libraries (JCDL)*, Jul. 2002, pp. 65–73.
[59] S. W. Jang, S. Kim, and J. Ha, "Graph-based recommendation systems: Comparison analysis between traditional clustering techniques and neural embedding," Stanford Univ., Stanford, CA, USA, Tech. Rep. 58, 2007.
[60] S. Chaudhari, A. Azaria, and T. Mitchell, "An entity graph based recommender system," *AI Commun.*, vol. 30, no. 2, pp. 141–149, May 2017.
[61] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[62] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2011, pp. 635–644.
[63] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, and A. Aspuru-Guzik, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.
[64] M. C. F. de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 3, pp. 378–394, Jul. 2003.
[65] L. vsan der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[66] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 701–710.
[67] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 1067–1077.
[68] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.
[69] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
[70] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907
[71] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*. [Online]. Available: http://arxiv.org/abs/1706.02263
[72] Q. Lu and L. Getoor, "Link-based classification," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 496–503.
[73] D. Koller, N. Friedman, S. Džeroski, C. Sutton, A. McCallum, A. Pfeffer, *Introduction to Statistical Relational Learning*. Cambridge, MA, USA: MIT Press, 2007.
[74] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proc. IEEE*, vol. 104, no. 1, pp. 11–33, Jan. 2016.
[75] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7370–7377.
[76] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 793–803.
[77] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang, "SimGNN: A neural network approach to fast graph similarity computation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 384–392.
[78] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167.

[79] O. Levy and Y. Goldberg, "Linguistic regularities in sparse and explicit word representations," in *Proc. 18th Conf. Comput. Natural Lang. Learn.*, 2014, pp. 746–751.
[80] D. Paranyushkin, "InfraNodus: Generating insight using text network analysis," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 3584–3589.
[81] M. Litvak, M. Last, and M. Friedman, "A new approach to improving multilingual summarization using a genetic algorithm," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 927–936.
[82] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of Web documents using graph matching," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 3, pp. 475–496, May 2004.
[83] A. H. Osman and N. Salim, "An improved semantic plagiarism detection scheme based on chi-squared automatic interaction detection," in *Proc. Int. Conf. Comput., Electr. Electron. Eng. (ICCEEE)*, Aug. 2013, pp. 640–647.
[84] A. H. Osman and O. M. Barukab, "SVM significant role selection method for improving semantic text plagiarism detection," *Int. J. Adv. Appl. Sci.*, vol. 4, no. 8, pp. 112–122, Aug. 2017.
[85] M. Mishra, J. Huan, S. Bleik, and M. Song, "Biomedical text categorization with concept graph representations using a controlled vocabulary," in *Proc. 11th Int. Workshop Data Mining Bioinf. (BIOKDD)*, 2012, pp. 26–32.
[86] S. Bleik, M. Mishra, J. Huan, and M. Song, "Text categorization of biomedical data sets using graph kernels and a controlled vocabulary," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 5, pp. 1211–1217, Sep. 2013.
[87] K. Bijari, H. Zare, E. Kebriaei, and H. Veisi, "Leveraging deep graph-based text representation for sentiment polarity applications," *Expert Syst. Appl.*, vol. 144, Apr. 2020, Art. no. 113090.
[88] M. Litvak and M. Last, "Graph-based keyword extraction for single-document summarization," in *Proc. Workshop Multi-Source Multilingual Inf. Extraction Summarization*, 2008, pp. 17–24.
[89] M. Crochemore and R. Vérin, "Direct construction of compact directed acyclic word graphs," in *Proc. Annu. Symp. Combinat. Pattern Matching*, 1997, pp. 116–129.
[90] N. Xu, A.-A. Liu, J. Liu, W. Nie, and Y. Su, "Scene graph captioner: Image captioning based on structural visual representation," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 477–485, Jan. 2019.
[91] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognit. Lett.*, vol. 18, no. 8, pp. 689–694, Aug. 1997.
[92] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognit. Lett.*, vol. 19, nos. 3–4, pp. 255–259, Mar. 1998.
[93] M.-L. Fernández and G. Valiente, "A graph distance metric combining maximum common subgraph and minimum common supergraph," *Pattern Recognit. Lett.*, vol. 22, nos. 6–7, pp. 753–758, May 2001.
[94] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray, "Graph distances using graph union," *Pattern Recognit. Lett.*, vol. 22, nos. 6–7, pp. 701–704, May 2001.
[95] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Graph representations for Web document clustering," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2003, pp. 935–942.
[96] A. Schenker, H. Bunke, M. Last, and A. Kandel, "A graph-based framework for Web document mining," in *Proc. Int. Workshop Document Anal. Syst.*, 2004, pp. 401–412.
[97] A. Schenker, H. Bunke, M. Last, and A. Kandel, "Clustering of Web documents using graph representations," in *Applied Graph Theory in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer, 2007, pp. 247–265.
[98] M. Nasr Azadani, N. Ghadiri, and E. Davoodijam, "Graph-based biomedical text summarization: An itemset mining and sentence clustering approach," *J. Biomed. Informat.*, vol. 84, pp. 42–58, Aug. 2018.
[99] M. Mesgar and M. Strube, "Graph-based coherence modeling for assessing readability," in *Proc. 4th Joint Conf. Lexical Comput. Semantics*, 2015, pp. 309–318.
[100] M. Johnson, "PCFG models of linguistic tree representations," *Comput. Linguistics*, vol. 24, no. 4, pp. 613–632, 1998.
[101] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 175–180.
[102] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[103] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[104] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," 2017, *arXiv:1703.04826*. [Online]. Available: http://arxiv.org/abs/1703.04826
[105] U. Sawant, S. Garg, S. Chakrabarti, and G. Ramakrishnan, "Neural architecture for question answering using a knowledge graph and Web corpus," *Inf. Retr. J.*, vol. 22, nos. 3–4, pp. 324–349, Aug. 2019.
[106] Y. Chen, J. Wang, P. Li, and P. Guo, "Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph," *Comput. Speech Lang.*, vol. 57, pp. 98–107, Sep. 2019.
[107] W. Fan, H. Liu, S. Wang, Y. Zhang, and Y. Chang, "Extracting keyphrases from research papers using word embeddings," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2019, pp. 54–67.
[108] Y. Zhang, H. Liu, S. Wang, W. H. Ip., W. Fan, and C. Xiao, "Automatic keyphrase extraction using word embeddings," *Soft Comput.*, vol. 24, no. 8, pp. 5593–5608, Apr. 2020.
[109] A. Duque, M. Stevenson, J. Martinez-Romo, and L. Araujo, "Co-occurrence graphs for word sense disambiguation in the biomedical domain," *Artif. Intell. Med.*, vol. 87, pp. 9–19, May 2018.
[110] R. Wang, H. Zhao, S. Ploux, B.-L. Lu, M. Utiyama, and E. Sumita, "Graph-based bilingual word embedding for statistical machine translation," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 17, no. 4, pp. 1–23, Aug. 2018.
[111] A. Khan, N. Salim, H. Farman, M. Khan, B. Jan, A. Ahmad, I. Ahmed, and A. Paul, "Abstractive text summarization based on improved semantic graph approach," *Int. J. Parallel Program.*, vol. 46, no. 5, pp. 992–1016, Oct. 2018.
[112] W. Shi, W. Zheng, J. X. Yu, H. Cheng, and L. Zou, "Keyphrase extraction using knowledge graphs," *Data Sci. Eng.*, vol. 2, no. 4, pp. 275–288, Dec. 2017.
[113] S. Jinarat, B. Manaskasemsak, and A. Rungsawang, "Short text clustering based on word semantic graph with word embedding model," in *Proc. Joint 10th Int. Conf. Soft Comput. Intell. Syst. (SCIS) 19th Int. Symp. Adv. Intell. Syst. (ISIS)*, Dec. 2018, pp. 1427–1432.
[114] E. Shabunina and G. Pasi, "A graph-based approach to ememes identification and tracking in social media streams," *Knowl.-Based Syst.*, vol. 139, pp. 108–118, Jan. 2018.
[115] M. Dutta, A. K. Das, C. Mallick, A. Sarkar, and A. K. Das, "A graph based approach on extractive summarization," in *Emerging Technologies in Data Mining and Information Security*. Singapore: Springer, 2019, pp. 179–187.
[116] J. Wu, Z. Xuan, and D. Pan, "Enhancing text representation for classification tasks with semantic graph structures," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 5, 2011.
[117] S. Bordag, G. Heyer, and U. Quasthoff, "Small worlds of concepts and other principles of semantic search," in *Proc. Int. Workshop Innov. Internet Community Syst.*, 2003, pp. 10–19.
[118] L. Ramachandran and E. F. Gehringer, "Determining degree of relevance of reviews using a graph-based text representation," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 442–445.
[119] B. Galitsky, D. Ilvovsky, S. O. Kuznetsov, and F. Strok, "Matching sets of parse trees for answering multi-sentence questions," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process. (RANLP)*, 2013, pp. 285–293.
[120] M. Ajgalik, M. Barla, and M. Bielikova, "From ambiguous words to key-concept extraction," in *Proc. 24th Int. Workshop Database Expert Syst. Appl.*, Aug. 2013, pp. 63–67.
[121] S. Hensman, "Construction of conceptual graph representation of texts," in *Proc. Student Res. Workshop HLT-NAACL XX-HLT-NAACL*, 2004, pp. 49–54.
[122] D. Kobayashi, T. Yoshikawa, and T. Furuhashi, "Visualization and analytical support of questionnaire free-texts data based on HK graph with concepts of words," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2011, pp. 1339–1343.
[123] M. Sigman and G. A. Cecchi, "Global organization of the wordnet lexicon," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 3, pp. 1742–1747, Feb. 2002.
[124] A. E. Motter, A. P. S. de Moura, Y.-C. Lai, and P. Dasgupta, "Topology of the conceptual network of language," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 6, Jun. 2002, Art. no. 065102.
[125] R. Blanco and C. Lioma, "Graph-based term weighting for information retrieval," *Inf. Retr.*, vol. 15, no. 1, pp. 54–92, Feb. 2012.

[126] S. S. Sonawane and P. A. Kulkarni, "Graph based representation and analysis of text document: A survey of techniques," *Int. J. Comput. Appl.*, vol. 96, no. 19, pp. 1–8, 2014.

[127] H. Bunke, "Error correcting graph matching: On the influence of the underlying cost function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 917–922, 1999.

[128] D. H. Kim, I. D. Yun, and S. U. Lee, "A comparative study on attributed relational gra matching algorithms for perceptual 3-D shape descriptor in MPEG-7," in *Proc. 12th Annu. ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2004, pp. 700–707.

[129] J. Gao and J. Gao, "A similarity measurement method based on graph kernel for disconnected graphs," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 6430–6431.

[130] B. Gallagher, "The state of the art in graph-based pattern matching," Lawrence Livermore Nat. Lab., Livermore, CA, USA, Tech. Rep. 895418, 2006.

[131] B. Gallagher, "Matching structure and semantics: A survey on graph-based pattern matching," in *Proc. AAAI Fall Symp. Capturing Using Patterns Evidence Detection*, 2006, pp. 45–53.

[132] J. R. Ullmann, "An algorithm for subgraph isomorphism," *J. ACM*, vol. 23, no. 1, pp. 31–42, Jan. 1976.

[133] B. T. Messmer and H. Bunke, "Subgraph isomorphism detection in polynomial time on preprocessed model graphs," in *Proc. Asian Conf. Comput. Vis.*, 1995, pp. 373–382.

[134] B. D. McKay, "Nauty user's guide (version 1.5)," Dept. Comput. Sci., Austral. Nat. Univ., Canberra, ACT, Australia, Tech. Rep. TR-CS-90-02, 1990.

[135] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM SIGKDD Explor. Newslett.*, vol. 5, no. 1, pp. 59–68, Jul. 2003.

[136] D. J. Cook, L. B. Holder, and S. Djoko, "Knowledge discovery from structural data," *J. Intell. Inf. Syst.*, vol. 5, pp. 229–248, 1995.

[137] R. C. Wilson and E. R. Hancock, "A Bayesian compatibility model for graph matching," *Pattern Recognit. Lett.*, vol. 17, no. 3, pp. 263–276, Mar. 1996.

[138] W.-H. Tsai and K.-S. Fu, "Error-correcting isomorphisms of attributed relational graphs for pattern analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 12, pp. 757–768, Dec. 1979.

[139] L. Cinque, D. Yasuda, L. G. Shapiro, S. Tanimoto, and B. Allen, "An improved algorithm for relational distance graph matching," *Pattern Recognit.*, vol. 29, no. 2, pp. 349–359, Feb. 1996.

[140] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 5, pp. 504–519, Sep. 1981.

[141] D. Shasha, J. T. L. Wang, and R. Giugno, "Algorithmics and applications of tree and graph searching," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. (PODS)*, 2002, pp. 39–52.

[142] N. Guarino, "Formal ontology, conceptual analysis and knowledge representation," *Int. J. Hum.-Comput. Stud.*, vol. 43, nos. 5–6, pp. 625–640, Nov. 1995.

[143] T. Coffman, S. Greenblatt, and S. Marcus, "Graph-based technologies for intelligence analysis," *Commun. ACM*, vol. 47, no. 3, pp. 45–47, Mar. 2004.

[144] B. Aleman-Meza, C. Halaschek-Wiener, I. B. Arpinar, C. Ramakrishnan, and A. P. Sheth, "Ranking complex relationships on the semantic Web," *IEEE Internet Comput.*, vol. 9, no. 3, pp. 37–44, May 2005.

[145] H. Bunke, X. Jiang, and A. Kandel, "On the minimum common supergraph of two graphs," *Comput.*, vol. 65, pp. 13–25, Jul. 2000.

[146] A. Berry and A. Sigayret, "Representing a concept lattice by a graph," *Discrete Appl. Math.*, vol. 144, nos. 1–2, pp. 27–42, 2004.

[147] M. Wolverto, I. Harrison, J. Lowrance, A. Rodriguez, and J. Thomere, "Software supported pattern development in intelligence analysis," in *Proc. IEEE Int. Conf. Comput. Intell. Homeland Secur. Pers. Saf.*, Oct. 2006, pp. 5–10.

[148] M. Kuramochi and G. Karypis, "Finding frequent patterns in a large sparse graph," *Data Mining Knowl. Discovery*, vol. 11, no. 3, pp. 243–271, 2005.

[149] Z. Sun, H. Wang, H. Wang, B. Shao, and J. Li, "Efficient subgraph matching on billion node graphs," *Proc. VLDB Endowment*, vol. 5, no. 9, pp. 788–799, May 2012.

[150] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1367–1372, Oct. 2004.

[151] T. Neumann and G. Weikum, "The RDF-3X engine for scalable management of RDF data," *VLDB J.*, vol. 19, no. 1, pp. 91–113, Feb. 2010.

[152] L. B. Holder, D. J. Cook, and S. Djoko, "Substucture discovery in the SUBDUE system," in *Proc. KDD Workshop*, 1994, pp. 169–180.

[153] M. Atre, V. Chaoji, M. J. Zaki, and J. A. Hendler, "Matrix 'bit' loaded: A scalable lightweight join query processor for RDF data," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 41–50.

[154] F. Zhu, Q. Qu, D. Lo, X. Yan, J. Han, and P. S. Yu, "Mining top-k large structural patterns in a massive network," *Proc. VLDB Endowment*, vol. 4, pp. 807–818, Aug. 2011.

[155] L. Zou, L. Chen, and M. T. Özsu, "Distance-join: Pattern match query in a large graph database," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 886–897, Aug. 2009.

[156] J. Cheng, J. X. Yu, B. Ding, P. S. Yu, and H. Wang, "Fast graph pattern matching," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 913–922.

[157] H. He and A. K. Singh, "Graphs-at-a-time: Query language and access methods for graph databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2008, pp. 405–418.

[158] S. Zhang, S. Li, and J. Yang, "GADDI: Distance index based subgraph matching in biological networks," in *Proc. 12th Int. Conf. Extending Database Technol. Adv. Database Technol. (EDBT)*, 2009, pp. 192–203.

[159] P. Zhao and J. Han, "On graph query optimization in large networks," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 340–351, Sep. 2010.

[160] X. Yan, P. S. Yu, and J. Han, "Graph indexing: A frequent structure-based approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data SIGMOD*, 2004, pp. 335–346.

[161] H. He and A. K. Singh, "Closure-tree: An index structure for graph queries," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, p. 38.

[162] J. Cheng, Y. Ke, W. Ng, and A. Lu, "FG-index: Towards verification-free query processing on graph databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2007, pp. 857–872.

[163] P. Zhao, J. X. Yu, and P. S. Yu, "Graph indexing: Tree+ delta>= graph," in *Proc. 33rd Int. Conf. Very Large Data Bases*, 2007, pp. 938–949.

[164] Y. Tian and J. M. Patel, "TALE: A tool for approximate large graph matching," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 963–972.

[165] H. Shang, Y. Zhang, X. Lin, and J. X. Yu, "Taming verification hardness: An efficient algorithm for testing subgraph isomorphism," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 364–375, Aug. 2008.

[166] M. Mongiovì, R. Di Natale, R. Giugno, A. Pulvirenti, A. Ferro, and R. Sharan, "SIGMA: A set-cover-based inexact graph matching algorithm," *J. Bioinf. Comput. Biol.*, vol. 08, no. 02, pp. 199–218, Apr. 2010.

[167] A. Khan, N. Li, X. Yan, Z. Guan, S. Chakraborty, and S. Tao, "Neighborhood based fast graph search in large networks," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2011, pp. 901–912.

[168] W.-S. Han, J. Lee, and J.-H. Lee, "Turbo$_{iso}$ towards ultrafast and robust subgraph isomorphism search in large graph databases," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 337–348.

[169] M. Saltz, A. Jain, A. Kothari, A. Fard, J. A. Miller, and L. Ramaswamy, "DualIso: An algorithm for subgraph pattern matching on very large labeled graphs," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2014, pp. 498–505.

[170] V. Carletti, P. Foggia, M. Vento, and V. Plus, "An improved version of VF2 for biological graphs," in *Proc. Conf. Graph-Based Represent. Pattern Recognit.* Beijing, China, 2015, pp. 168–177.

[171] C. Nabti and H. Seba, "Subgraph isomorphism search in massive graph databases," in *Proc. Int. Conf. Internet Things Big Data-IoTBD*, 2016, pp. 204–213.

[172] M. Asiler and A. Yazıcı, "BB-graph: A subgraph isomorphism algorithm for efficiently querying big graph databases," 2017, *arXiv:1706.06654*. [Online]. Available: http://arxiv.org/abs/1706.06654

[173] V. Carletti, P. Foggia, A. Saggese, and M. Vento, "Introducing VF3: A new algorithm for subgraph isomorphism," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit.*, 2017, pp. 128–139.

[174] L. Zhu, Y. Yao, Y. Wang, X. Hei, Q. Zhao, W. Ji, and Q. Yao, "A novel subgraph querying method based on paths and spectra," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 5671–5678, Sep. 2019.

[175] T. Ma, S. Yu, J. Cao, Y. Tian, and M. Al-Rodhann, "InfMatch: Finding isomorphism subgraph on a big target graph based on the importance of vertex," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121278.

AHMED HAMZA OSMAN received the bachelor's degree in computer science from the International University of Africa, the master's degree in computer science from the Sudan University of Science and Technology, Sudan, and the Ph.D. degree (Hons.) in computer science from Universiti Teknologi Malaysia (UTM).

He was the Head of Computer Science Department, Faculty of Computer Studies, International University of Africa. He currently works as an Associate Professor with King Abdulaziz University (KAU), Saudi Arabia. His research interests include information retrieval, plagiarism detection, soft computing, data mining, natural language processing, and text summarization.

OMAR MOHAMMED BARUKUB was born in Al-Taif, Saudi Arabia. He received the B.Sc. degree from Electrical Engineering and Computer Department, College of Engineering, King Abdulaziz University (KAU), Jeddah, in 1987, and the M.Sc. degree in information technology and the Ph.D. degree in computer engineering from the College of Engineering, Florida Institute of Technology, in 1999.

From 1999 to 2011, he was working at the College of Telecom and Electronics, Saudi Arabia, he was appointed as an Associate Professor with the Faculty of Computing and Information Technology, KAU, Rabigh, from 2011 to 2016, where he is currently a Full Professor and the Dean. His research interests include logic-modal logic, mobile agent, cryptography, data mining, information security, and audit.